An Interface for 3D Content for Cultural Heritage Applications

Thierry Zoghbi

A thesis submitted for the degree of
Master of Science
at the University of East Anglia
February 2010

An Interface for 3D Content for Cultural Heritage Applications

Thierry Zoghbi

Acknowledgements

I would like to take this moment to thank the following people who helped make this Thesis a success.

Prof. Andy Day (CMP, UEA) & Dr. Stephen Laycock (CMP, UEA) for their continued support and advice throughout the course of my research and write up.

Mr. David Drinkwater (UMG, UEA) for 3D models of St Andews Hall, including textures and relevant historical books and maps.

Prof. Carole Rawcliffe (HIS, UEA) & Dr. Christopher Bonfield (HIS, UEA) for their expert historical input as well as historical documentation.

Finally to my Parents, Elie and Mona for their emotional and financial support whom without, these past years would not have been possible.

Table of Contents

List of Tables

List of Figures

Chapter 1

Background

Virtual Reality (VR) is playing an ever increasing role in virtual museum guides [LSB+04] [KRB06], historical and cultural reconstruction and preservation, as well as education for people of all ages. As computers and high speed internet connections are more readily available to the public, it is no longer necessary to take a trip down to the local museum to have a look at artefacts and learn about them. Archaeologists currently have the ability to view the past in 3D, with the help of virtual heritage sites, that show what buildings and areas used to look like at certain periods of time. Now it is even possible to have augmented reality, where virtual reality allows one to create full scale 3D animations and simulations of the real world, this also allows one to superimpose layers of additional information [Vel99] on top of real world scenes. Virtual museums and libraries create digital versions of their physical spaces. On July 9th, 1991, one of the earliest experiments with such systems was the Micro-Gallery at the National Gallery of England (London) - this contained a small room of computers within the physical gallery, where visitors could view images of paintings in the collection in keeping with their personal interest [Rub92]. Virtual libraries and museums can vary in complexity, a number of museums include Quick-Time Virtual Reality tours on CDs. Other museums such as the NHM (Natural History Museum)

London, have a virtual wonders area, which contains a collection of strange and interesting virtual objects, most of which are not found in the public galleries. Some virtual visits can go further by simply visiting the rooms of the museum ahead of time. In a project by the Urban Modelling Group for Norwich Blackfriars Hall [UMG08], portrait subjects jump out of their frames in 3D and start walking around a virtual room whilst "talking" about their history.

Figure 1.1: Screen Shot of Pettus - St Andrews Hall, Norwich. Urban Modelling Group.

Museums and galleries have traditionally been famous for their "do not touch" signs. However many visitors especially children want to know how things feel. This is now an area where VR reconstruction of objects are linked with haptic feedback devices allowing people to *feel* the object as though it were real [LLD06]. As explained VR allows complete reconstructions of objects, archaeological sites and historical monuments in three-dimensions.

Some very good examples of VR reconstructions are being created by infobyte (Rome) [inf]. These include reconstructions of the *Upper Church of San Francesco* (Assisi) and *Saint Peter's Basilica*(Vatican).

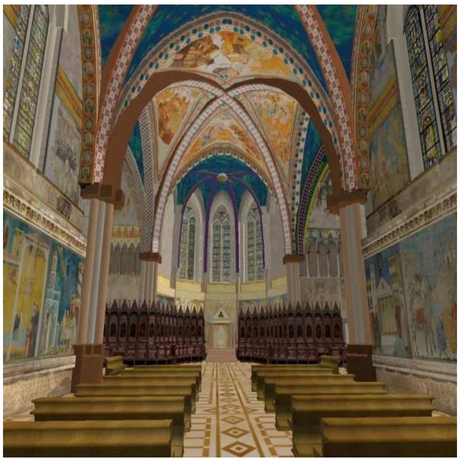

Figure 1.2: Screenshot of La Basilica Superiore [bas09]

The principles of historical virtual museums is also being applied to urban landscapes to create historical virtual cities. For example, CINCEA, as part of the MOSAIC project [Bor08] has reconstructed the medieval city of Bologna. In such cases reconstructions of cultural heritage become significant for social and even economic history; meaning that people will be able to look back and get an idea of how well an area, city or even individual races where doing for themselves, by looking at the quality and grandeur of their buildings and urban environments.

One of the first pioneering interface projects conducted by Negroponte's architecture machine group at MIT, was the 'put that there' interface [amgaM10]. The project consisted of voice and gesture commands to manipulate simple shapes about a large-screen graphics display surface [Bol80].

Chapter 2

Introduction

This chapter looks into the motivations and research behind this thesis as well as providing a terminology section which aims to explain technical terms used in this document.

The thesis will investigate and discuss current cultural heritage sites and applications, as well as looking into different ideas and techniques in which user interaction can be easily integrated to aid in the education of users of all ages, as well as the preservation and reconstruction of the heritage site itself. The application will be constructed using Flash Actionscript 3 to allow seamless integration into websites in order to reach a wider audience.

The author aims to re-model the Cloisters of St Andrew's Hall, Norwich, and the surrounding area as it would have looked in the early 18th century. St. Andrews and Blackfriars Hall once formed part of a larger church, the great conventual church of St John the Baptist, built by the Dominican Friars (Black Friars) and was finished in 1471 after a fire had destroyed the original building in 1413. According to a green street plaque put up by the city, St. Andrew's Hall was built at the expense of Sir Thomas Erpingham. Exterior and interior animations of the buildings and surrounding areas will be rendered and converted into short walkthroughs, which will be integrated into the flash based application to simulate the illusion of real time environments. The user interface that will be created will seem to allow the user to navigate through different areas at will, whilst providing useful facts and educational information about the area through which they are travelling.

2.1 Motivations and Research Objectives

Although computer systems are now generally powerful enough to handle real time graphics, and are readily available to the public, it is still difficult to have high polygon (photo-realistic) models streaming over the internet in real time, whilst giving the user the ability to navigate around a 3D world. However recent open source Flash repositories have made this possible by making use of Papervision3D code, (see §7.2).

Navigating around a 3D environment using 2D inputs (keyboard and mouse) can become very tricky and confusing. This is where a user friendly HCI will be developed in order to aid the user in the navigation of the virtual world. The aim is to create a design that will allow people of all ages to experience historical buildings and their "artifacts" within the comfort of their own home. This could also be used as an educational tool within the class room, as well as an informative guide for visitors and tourists to museums and heritage sites.

Primarily the system designed is to be incorporated into websites, and the illusion of real time animations is achieved.

2.1.1 Research approaches

Initially, all buildings were modelled from photographs of the building that still exist, as well as historical images and maps obtained from the UEA History Department; an existing model of St Andrews Hall was obtained from David Drinkwater [UMG08] and used as part of the overall 3D model. Research into current techniques was undertaken to see what technology already existed, in order to develop an interface utilising the necessary software and designs to best fit the aims of this thesis. Once that was completed, 3D was incorporated into the interface allowing for a degree of interactivity to aid user enjoyability.

2.2 Thesis Outline

This thesis is organised as follows:

Chapter 3 will detail the existing approaches for the delivery of 3D content on the web, with a particular focus on cultural heritage applications.

Chapter 4 discusses the historical aspect of St. Andrews Hall which will give the reader some background into the 3D development of the building and surrounding area.

Chapter 4.3 shows the resources used to base the 3D modelling on, so that it was historically accurate, and will aim to describe several techniques used in the modelling process.

Chapter 5 describes different types of input devices and user interaction. It also talks about interfaces from a users point of view, in regards to best practices, different types of controls and the principles involved in achieving a good user interface for the finest user experience possible.

Chapter 6 will talk about video encoding and the flash video player, as well as looking into different third party software that allows 3D user interaction in a flash environment, whilst discussing some of the problems that would be encountered.

The author aims to show the stages of program development in Chapter 7 as well as important areas of coding that made the program achievable, including results of the experiments conducted to test the application.

Chapter 9 concludes the topics discussed in this thesis, as well as looking into possible research expandability.

2.3 Terminology

2.3.1 CAVE

The CAVE Automatic Virtual Environment [Fri99] and supporting software the CAVE libraries (CAVElib) were developed at the Electronic Visualization Laboratory (EVL) in Chicago, and introduced at SIGGRAPH '92. One of the design goals was the ability to mix real and virtual objects and the presence of scale. This is achieved automatically since the user's body is visible inside the virtual environment as an avatar. It is a room-sized cubic system with projections on several walls and the floor that aim to simulate the user being in a real cave environment.

2.3.2 DOF

In the context of haptic feedback devices this quantity is used to describe how many translations and rotations are utilised by a device. A haptic mouse for instance moves in the x-y plane and so has two degrees of freedom (DOF). Three-dimensional devices are able to translate along the x-axis, y-axis and z-axis, and are classed as three DOF. Three-dimensional devices, which are also able to rotate about all three axes, have six DOF. Arm and hand haptic devices can have more degrees of freedom, as the force feedback can be applied to other points such as the forearm and wrist.

2.3.3 Semi-Realtime

Semi-realtime is a term fabricated by the author, in order to describe the ability of a user appearing to navigate freely within a virtual environment, whilst being constrained by pre-rendered areas.

Chapter 3

Cultural Heritage and Historical Reconstruction

There are many current applications in which virtual heritage sites have been applied to the web (see §3.1.1) in favour of conservation and education. Not only can digital technologies be used for a virtual reconstruction of architectural monuments, but they can also be used in support of simulation techniques and the scientific work of historians, art historians, archaeologists and restorers [Enc99]. According to commentaries in the press [Enc99] approximately 1.65million people visited the website of the museum of modern art (New York) in 1997, for a virtual tour. One can only assume that this number would now be substantially larger as more people are online. This shows that there is a need for virtual sites and they seem to be becoming ever increasingly popular [MFMR09]. Cultural heritage is becoming an important application for VR technology. An EC/NSF (European Commission and National Science Foundation) Advanced Research Workshop identified it as one of the key application domains for driving the development of new HCIs [BvDE+99], [EV01] and [LD08]. There have been a number of research projects that have looked into developing accurate digital reconstructions of ancient artefacts. For example Boulanger et al. [BTEHM98] describe the work done in creating models of the tombs of Nefertari and Tutankhamun, and presenting a VR tour of the models in a public museum.

3.1 Current applications for Heritage sites

3.1.1 Virtual Pompeii

The Virtual Pompeii project was developed in order to reconstruct the theatre district of the ancient Roman city of Pompeii, this included three dimensional models of the Temple of Isis, the Grand Theatre, the Triangular Forum and connecting areas. This area was specifically chosen because it was extensively documented and contained a variety of activities that provided a useful cross reference of Roman urban life [JV05]. In 2005 the Virtual Pompeii project was completely open to the public online, it was used to encourage educators, developers and students alike to develop educational applications based wholly or in part on the materials used to create the Pompeii project. The idea was not just to be able to see the materials used, but to download, enhance and alter them for educational use [Dis09]. Virtual Pompeii was created in VRML (§6.3.3), which was widely used in educational websites containing simulations of ancient architecture. There are many internet technologies that exist that are technically superior to VRML [Con08] which allow 3D modellers to import objects from the VRML environment and convert them into their preferred format (§6.3).

Figure 3.1: Inside the Virtual Temple of Isis [J.J99]

3.1.2 Google Earth: Rome Reborn

Google have recently launched a 3D version of Ancient Rome based on its digital re-creation by the University of Virginia's Institute for Advanced Technology in the Humanities, on its Google Earth application [rom08c]. The user is able to navigate around the city and see exactly what it would have looked like in 320 A.D. which was more or less the height of its development as the capital of the Roman Empire. Their secondary, but important aim was to create the cyber-infrastructure whereby the model could be updated and corrected. The knowledge known about the city was used to correctly reconstruct the topography and urban infrastructure and wherever possible sources of archaeological information would be made available to the public [Rom08b]. Not only were the Exteriors of the building accurately modelled but the most important ones contained full detailed interiors, that the user could also walk and fly through. Figure 3.2 shows a screenshot of 3D Ancient Rome, before it was applied to the Google Earth application.

Figure 3.2: An aerial view of the entire city model - Rome Reborn [rom08a]

The Rome Reborn project showcases new approaches for exhibiting historical findings in museums, classrooms and on the internet. It allows users to explore and understand the past in different ways, by utilising technologies that are defining the future of computer graphics and interactive techniques. The entire project could not have been achieved had it not been for the collaboration between computer scientists and humanists inside several universities and partnerships in industry [fATitH08].

City Engine

Manually modelling and rebuilding a 3D city is very time consuming on a project as big as Rome Reborn, it would have been a colossal feat had the project team not made use of procedural modelling techniques [Pro09a](see §3.2.3). Procedural's CityEngine [cit09] was the software chosen to utilise this form of technology. It is based on L-systems that rely on various image maps such as land-water boundaries and population density that allow it to generate a system of roads and vegetation, divide them into lots and create appropriate geometry for buildings [PM01], as well as incorporating parametric modelling, which allows the user to quickly adjust various constants such as height and size while keeping all architectural elements correctly aligned [Pro09b].

3.1.3 Reconstruction of Braga's Cathedral

VirtualBraga's Cathedral was developed by Centro de Computao Grfica (CCG) in Coimbra in cooperation with Unidade de Arqueologia da Universidade do Minho-UAUM (Portugal). It was a multimedia kiosk which showed the cathedral of Braga together with its surrounding archaeological excavations [Vil09]. Each of the excavation fields is introduced by a short text description, photos and drawings, through to a virtual model that enabled virtual walkthroughs. The overall aim of the project was to simulate interest for sponsors and to raise financial funding.

3.1.4 Dunhuang Magao Cave 322

The Magao caves in Dunhuang, China, date from as early as 618AD and are restricted to guided tours only, operate a strict policy on prohibiting photography as well as closures due to bad weather types and inaccessible facilities for the disabled [Aca10]. Recently ARC3D an Automatic image based 3D reconstruction [Con10] developed by the VISICS lab at University of Leuven, was used to create a virtual model of the Mogao Cave 322 which can be viewed on the 3D Coform website [Cof10]. It demonstrates how museums and cultural monuments can be presented virtually on the internet and available for all people to access at their own comfort.

3.1.5 European Research

There have been many projects funded by the European Union for the research and development of 3D Cultural Heritage. MOSAIC (Museum Over States and virtual Culture), was supported by the European Union and aims to make available documents, images and artefacts that were hidden in archives of museums and libraries or in private rooms to scientists, students and all interested people. At the end of 1998, seven partners of the project have presented a feasibility study which was showing the way to a European virtual museum [Bor08]. The EPOCH (European Network of Excellence in Open Cultural Heritage) network is another EU project that is a combination of about a hundred European cultural institutions, working together to improve the quality and technology for Cultural Heritage [EPO09] It was set up in 2004 [Tec10] and continues to produce up to date yearly publications. More recent projects include 3D-COFORM, who "aim to establish 3D documentation as an affordable, practical and effective mechanism for long term documentation of tangible cultural heritage" [Cof10] and continue to support the VAST International Symposium on Virtual Reality, Archeology and Cultural Heritage [20110]. Virtual Past is a University of East Anglia Enterprise which bring together VR animators and GIS modellers to create historically accurate interactive websites and computer models of use in the heritage industry and education, this is designed to create interest and accessibility for historical buildings, visitor attractions or educational centres [Pas09], an example of such work are the 3D images and animations of Blackfriars hall (now known as St Andrews Hall) from the early 16th century.

3.1.6 CAVE: Virtual Harlem

The goal of the Harlem project was to provide an environment that contextualises the study of the Harlem Renaissance for students through the construction of a virtual reality scenario that represents Harlem, New York, as it existed between 1920 and 1930. Networked students could navigate through the streets and buildings of Harlem and see the shops, homes, theatres, churches and businesses that the period offered. It was also possible to watch dances of the period, listen to music of the time, and even listen to political speeches. The main features of the Harlem environment that were implemented for iGrid 2000 were a period trolley car for transportation and some basic virtual characters [PAC+01].

Figure 3.3: Virtual Harlem Demonstrated using CAVE

As the CAVE is networked, an instructor can be in a different place, taking students through the environment and answering questions in real time [PLJ+01].

3.1.7 St. Peter And St. Paul's Basilicas

St. Peter's Basilica [pet09] and St. Paul's Basilica [pau09] are in two different places, and their virtual tours vary from a walk through video with description to 360 degree panoramic animations of different areas respectively. St. Peter's Basilica has an introductory page which asks the user to begin the tour, this starts with a short video which then leads onto a split page. At the bottom there is a birds' eye view map of the area, with a pointer denoting the users current position. Displayed above that are panoramic images of the rooms, however buttons are embedded on top of these images which allow the user to navigate around the area from one room to another, or to bring up detailed information about selected artefacts.

St. Paul's virtual tour (Fig 3.4) is centred around a birds eye view of the building with areas that the user can select. This then loads a very high resolution panoramic image of the selected area, which the user can rotate using the computer mouse pointer. No other form of user interaction is provided.

3.2 Different Reconstruction techniques of historical and urban cities

There are currently many techniques that can be used to reconstruct and regenerate ancient and cultural buildings and cities. This section will discuss a few of these technologies. Where possible, emphasis is paid to the visualisation approaches for a wide audience, i.e. techniques targeted to the web.

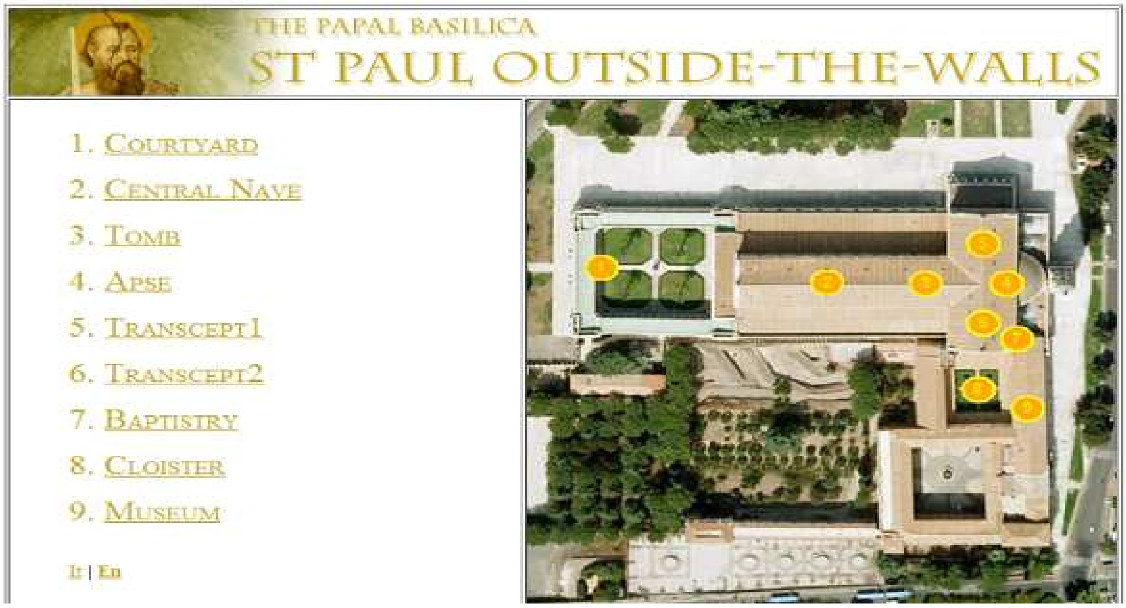

Figure 3.4: St. Paul's Basilica VR tour Website [pau09]

3.2.1 Image Based Reconstruction

The most basic image based reconstruction stems from the images of 2D exhibits such as painting or pictures of 3D objects, these are known as virtual image galleries. The advancement in technology to this technique are panoramic images that allow a visitor to have a full 360 degree view of an area or room, where not only is the exhibit captured but the space in which the exhibit is stored is also shown [Zar04].

3.2.2 3D Representation

While §3.2.1 discusses the uses of photo and video technology, it is not sufficient enough for a true virtual 3D environment. As intuitive people, humans prefer to see and feel objects from all possible angles, VRML allows a 3D mesh of triangles to be covered in textures which can be viewed in a web browser using VRML international standard [iso97]. Another example of presenting 3D environments online is using Flash (see Fig 3.5), an example of a collection of several 3D environments [nor09a] includes the Norwich Blackfriars site.

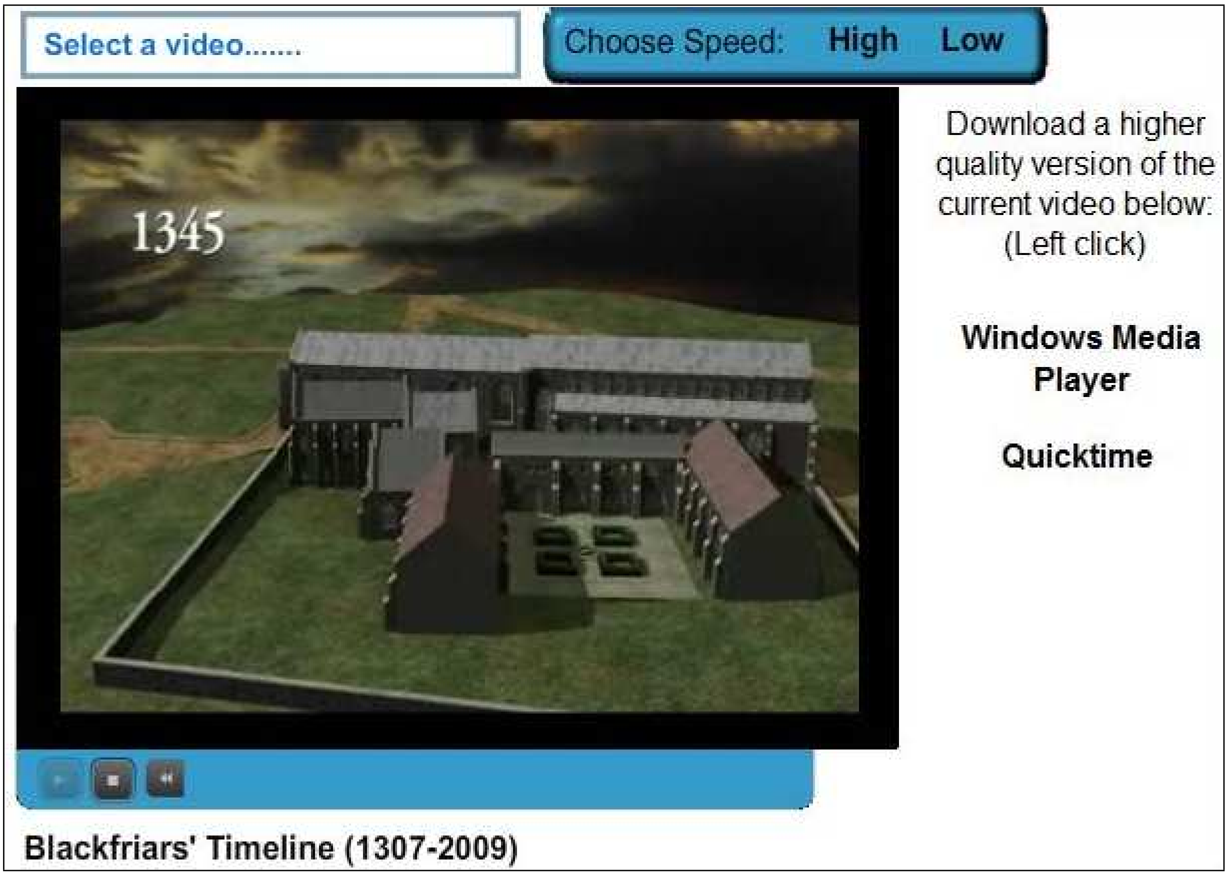

Figure 3.5: Screenshot of the Norwich Blackfriars Online System

However, it is costly in terms of data size and bandwidth used when delivered over the internet. One technique researched to reduce this problem as much as possible, is to allow the scene to be rendered jointly by the server and the client, in order to reduce the network requirements as much as possible [MCO97].

3.2.3 Procedural modelling

The creation of complex models is a crucial task in the development of computer games and movies with special effects. However, modelling large three-dimensional environments such as cities, requires a lot of time and can be a very expensive process. Procedural modelling is a system developed, that uses shape grammars capable of efficiently creating large cities with high geometric details up to a billion polygons [MWH+06] where necessary. For the construction of buildings the production rules use an iterative process that evolve a design by creating more and more details known as Progressive Refinement. For example, firstly a crude volumetric model of a building is created, then the façade is structured and then finally windows, doors and ornaments are added [MWH+06]. The idea of modelling environments using shape grammar was originally explored by George Stiny and James Gips in the 1970's [SG72] and then later by [PM01] in 2001. This procedural approach was based on L-systems to model cities, this system was called CityEngine. Most of the input data to build up the virtual city is represented by 2D image maps which control the behaviour of the system. For the creation of a complete city two different L-Systems are invoked, the first for building generation, and the second for street generation.

As well as procedural modelling for buildings, a similar technique can be utilised for modelling and rendering populated urban scenes. The CHARISMATIC project [BFW+01] is about the economic production and the interactive rendering of complete reconstructions of populated historical and cultural interest environments, often referred to as Cultural Heritage Sites [BWW+01], [WWDA01], [FWB+01]. The CHARISMATIC project designed a toolkit specifically to support the construction of these populated urban environments; with specially designed tools to model buildings and trees through to avatars and generic 3D objects. Examples of alternative similar technologies include [AADH+08] which aim to improve and release frameworks that support the incorporation of avatars in interactive real-time 3D VR systems. [DAHF04] discuss the way in which polygonal and multi-resolution surface techniques can complement one another in the modelling of urban environments. To simplify the user interaction, guide the operator and to create models that build in optimisations when attempting real-time rendering. [EGA+05] describes how a medieval European town from Lower Saxony (Germany) called Wolfenbttel, was modelled and populated with interactive multilingual avatars.

3.3 Comparison of current HCI for virtual heritage applications

This section will concentrate solely on the HCI and GUI look and feel of a few current virtual heritage applications. The author will briefly talk about how each application was laid out, personal opinions of how easy they are to use, and how any ideas can be incorporated into this project.

3.3.1 Virtual Old Prague

The VOP (Virtual Old Prague) application [old08] offers a smooth tour to selected regions of the real city, utilising a number of standard and modern technologies including VRML, Java and PHP. It is a web application that allows walking through a virtual city stored in a remote database a technique known as incremental transmission. The data is progressively transferred from the server according to the visitor's position. A high frame rate is achieved by rendering only visible city parts [Zar02]. Figure 3.6 gives an example of a simple layout of the VOP application.

3.3.2 Virtual Museum of the Ancient via Flaminia

The Virtual Museum of the Ancient via Flaminia contained a very elaborate website with a lot of Flash incorporated into it [fla08]. It was necessary for users to point their mouse to certain images or icons and wait for a text description. Although it was visually appealing it may not have been as straightforward to the entire demographic of users accessing the website. However after finding the virtual area of the website, users could sit back and watch the small animations and read relevant information to the right of the screen. This seems very helpful as it gives users the ability to

Figure 3.6: Virtual Old Prague application layout

learn about different artifacts without having to worry about navigation, but this does take away from any interactivity that a user may want to have, such as pausing the animation. Fig 3.7 shows the layout of virtual area on the Ancient via Flaminia website.

Figure 3.7: Ancient Flaminia application layout

3.3.3 The Great Hospital

Fig 3.8 shows the basic layout of The Great Hospital application, at first glance, it seems to be very similar in looks to what this project aims to achieve [Wil08]. However more detail will be emphasised on the user interface where users have more control over the 3D animation taking place. Rather than having the generic “play” and “stop” button that you generally see at the bottom of video players, this project aims to allow the user to actually choose which path they take from within the animation video itself.

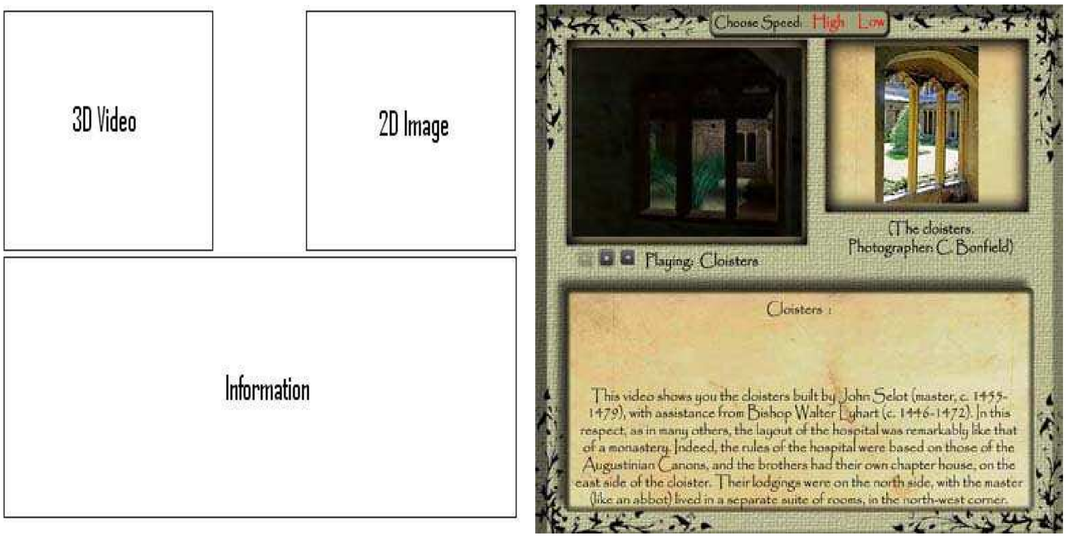

Figure 3.8: The Great Hospital application layout

3.3.4 Virtual Amsterdam Crystal Palace

Amsterdams Crystal Palace was built in 1864 but was destroyed by a fire in 1929 [cry08]. Before users are able to view and navigate around the 3D palace they must download and install a browser plugin called Blink 3D - this can become a little confusing for people who are not computer literate. The Virtual 3D palace navigation is very similar to that of Google Earth as mentioned in section 3.1.2. The entire screen is dedicated to the environment, with the user being expected to learn the different navigational keys and shortcuts. In the top left hand corner of the screen lies the command window, which gives the user feedback on shortcuts that have been applied. As this application is based around a real time environment, the bottom left of the screen contains information such as the FPS (frames per second). It also contains a chat window which would allow users to communicate with each other while in the environment. Figure 3.9 gives an idea of the applications layout.

Figure 3.9: Virtual 3D palace application layout

3.3.5 Conclusion of comparisons

In the authors opinion and for the purpose of this project, it was found that the best suited layout and user interface was the one used for the VOP project, as seen in figure 3.6. The design incorporated many different areas on a single page that allowed for different types of technologies and coding languages to be used. Although the VOP project environment loaded models using VRML, there is no reason why newer technologies can not replace it, such as Adobe Flash and Papervision 3D. Table 3.1 shows the basic comparisons between pre-rendered movies and Flash in real-time.

Comparisons	**Pre-Rendered Movies**	**Real time Flash / PV3D**
File Sizes	Generally large file sizes	Smaller Files sizes, better compression
Ability to stream online?	Yes	Yes
Interactive?	No	Highly Customisable and Interactive
Handle high polygon count?	Yes	Up to 20K depending on machine specification
Multi/ Cross-Platform compatible?	May require additional software players and Video Codecs	Yes
multi-browser compatibility?	plugins needed, may be difficult to find	Very common plugin, easy to install

Table 3.1: Table Comparing Pre-Rendered Movies and Real time Flash

Both techniques have shown to have their own advantages and disadvantages when it comes to providing multimedia content for the internet. It was for that reason the author decided to incorporate semi-realtime animations and augmented reality into the St Andrews Hall application. The idea behind semi-realtime Flash animations occurred initially because the author wanted to maximise the level of interactivity available, while keeping polygon detail high and file sizes low. This would allow quick loading times over basic broadband connections. The application was executed by first creating highly detailed 3D models of the required buildings, creating pre-rendered animations, then importing them into the Flash environment where it would be possible to compress file sizes and add layers of code seamlessly on top of the animations to create interactivity for the user.

Many aims of this thesis fit with some of the methods that were discussed in §3.1. For example the idea that a multimedia kiosk was used to bring together the Braga Cathedral, as well as archaeological excavations, along with short text descriptions through a virtual model, is similar to what has been implemented in this thesis.

Chapter 4

History and Modelling of St. Andrews Hall

There are some parts of Norwich Blackfriars (also known as St Andrews Hall) and its interior that no longer survive or have changed so considerably that it is impossible to recreate them exactly. In such cases experts have been consulted who have drawn on surviving evidence from other institutions, to provide an impression of how the building would have appeared during its heyday.

4.1 History

Although many of the buildings have been altered during their seven centuries of existence, the halls represent the most complete Friary church and Convent buildings surviving in the country [Sut77]. The buildings are a product of revolution in religious thinking, which began in southern Europe in the late 12th century as a reaction to the great wealth and temporal power acquired by some of the older religious orders.

St Dominic, born in Spain in 1170 [NA09], obtained papal permission in 1218 to create a new Order based on the rule of St. Benedict and called it the "Order of the Friars Preachers" In 1220 before he died, his last decision was to send a group of his followers to England, where the order was very successful and spread to Norwich in 1226.

The Dominicans, also known by the colour of their habit Black Friars, were one of the five friary orders to make their home in the city of Norwich.

A fire badly damaged the church and conventional buildings in 1413, which forced the Friars to move to their original quarters across the river, whilst they were re-building the structure.

After the Dissolution of the Monasteries, Augustine Steward (three times Mayor of Norwich) sent a proposition to Henry VIII in London, asking if the City of Norwich could buy the Dominican friary buildings [onl09]. By pledging to use the Halls for the good of the citizens, for fairs and feasting, Steward ensured the friary's survival. The use of the building considerably changed; the nave church was converted to an assembly hall and held a wide variety of functions. The infirmary over the south west corner of the cloister was probably used as a school, but this was soon moved. Most of the cloister ranges were used as granaries, and later in the late 17th century the buildings were used as Workhouses by the Guardians for the Poor. The north side of the cloister was used to examine and seal the cloths produced by the Walloon weavers. The cloisters were also used as building stores for the most of the 16th and 17th century [HH09]. The Norwich Mint was also operated for a time in one corner "259,000 in half-crowns, shillings and sixpences were minted in a corner of the Cloisters and these bear the letter N under the bust of William III." [Nor09b].

The main building to the north of the cloister, the Brewery, was the most profitable, being rented out at £25 per annum in 1603 (roughly £3600 in todays currency).

4.2 Guide to St Andrews Hall

St Andrews Hall is split up into many different sections. Main rooms include the Nave (St Andrews Hall), the Chancel (Blackfriars Hall) the Cloisters, Beckets Chapel and the Crypt. This section will provide a quick guide to the buildings:

The Plan:

In the 14th century, the English orders of the Friars had developed a unique method for laying out their buildings. The characteristics of these buildings were a large aisled nave used for preaching to big congregations and a smaller chancel without aisles, where the friars held their own services. The two areas were separated by a walkway between the nave and chancel, which gave direct access from the outside world to the cloister.

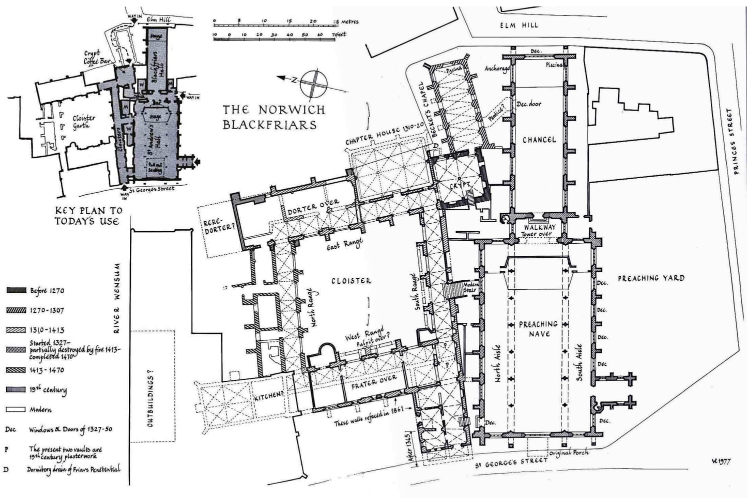

Figure 4.1: The Norwich Blackfriars Map

The Exterior of the Church:

The entire building as it stands today still contains the basic structure that began about 1327, when the Blackfriars first obtained the whole of this site and began building the east end, which was not completed until the closure of the lane in 1345. The Fire of 1413, destroyed the rood and most of the windows tracery.

The Nave Interior: The nave, now known as St Andrews hall lies to the left of the Friary church, it was probably finished about 1449, when the Friars re-occupied the site, but still retains some traces of the original church destroyed by the fire in 1413. The Friars stuck to their rule of simplicity in architecture as the nave is impressive for its size rather than any form of elaborate decoration. The Nave can be compared with the church of St Peter Mancroft on the Market Place constructed in similar decades between 1440-1460, this is because both buildings contained the same type of roof structure - the hammer beam.

Figure 4.2: The Nave Interior as it stands today.

The Chancel: Now known as Blackfriars' Hall, this was the Friars private chapel, and probably was not open to townspeople. As usual in Friaries there are no aisles to the chancel so that the great perpendicular windows are unusually high. One memorial remains from the long period during which this building housed the worship of the Dutch community in Norwich.

The Crypt and Beckets Chapel: This part of the building contains the oldest work on site, and has undergone many changes through medieval and post medieval times. The Crypt was first used as a chamber accommodation for the Friars Penitential, and was built between 1258 and 1267. The walls of the Crypt were made of brick, and must be one of the earliest building in the country to use this material. In

1271 the building was extended eastwards, and a vault on the crypt was built to bear an upper floor, as the Friars Penitential obtained permission from the rector of the a nearby church to celebrate services, ring their bell and bury the dead. The Crypt is now being used as a coffee bar, however the basic structure of the room remains untouched.

The Cloister: Till this day there have been no traces of the cloisters of the Friars Penitential, but three sides of the Dominican cloister survive, including portrait heads on the capitals towards the west end, which may well represent some of the early donors to the building. The arches between the cloister walk and the garth are examples of moulding brickwork. The 1st and 2nd bays north of the south-east corner show signs of the intersecting tracery which once filled the windows.

Figure 4.3: Exterior of the East Cloister, St Andrews Hall.

4.3 Modelling the Cloisters

In this section the author presents the modelling, texturing, lighting and rendering of the Cloisters and surrounding area with a view to create a set of high quality animations which can be both used online and as standalone animations. The focus in this section will be mainly on special techniques used to give distinctive effects for

Figure 4.4: Interior of the South Cloister, showing original brickwork.

the final renders and animations, and methods that were most useful while modelling. Throughout the course of the modelling, the work was coordinated with Prof. Carole Rawcliffe and Dr. Christopher Bonfield.(from the School of History, UEA) who are experts in medieval history.

4.3.1 Modelling and historical resources

Most of the structural information was attained from previous models of St Andrews Hall and the Cloisters in earlier periods made by David Drinkwater (UMG). Floor plans of the site were also aquired from Prof. Rawcliffe (Fig 4.1 and Fig 4.5). Visits to the current site where the South side of the Cloister Range still stands allowed the author to take pictures, as it was understood that the structure of the building had not been altered since the 18th century. References from very similar buildings such as the Cathedral in Norwich (Fig 4.6 and Fig 4.7) were also taken.

Most of the site however remained fairly undocumented in the 18th century, and therefore with some guidance from Prof. Rawcliffe and Dr. Bonfield 'artistic license' was given in order to complete the model. For the grounds surrounding the site, contour maps of present day Norwich were attained, and then edited to fit in with the earlier period of the model.

Fig 4.5 was correctly scaled and imported into 3Ds Max onto a plane, where the buildings were modelled on top, in order to achieve the correct placement and

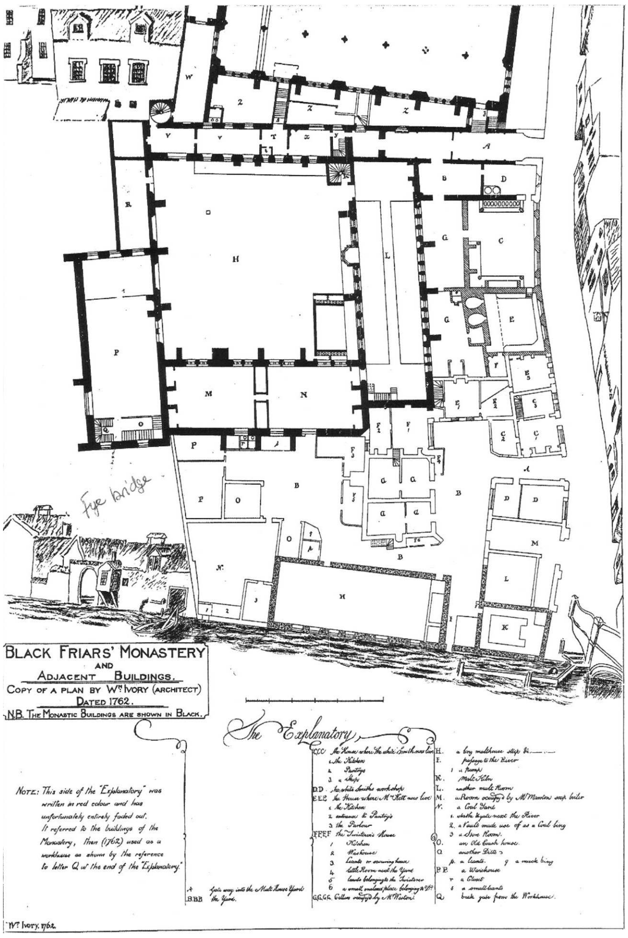

Figure 4.5: St Andrews and Surrounding Area in 1762

direction for the buildings in the area. Using such pictures as those shown in Fig 4.6 and 4.7 as well as following basic plans and estimating height elevations from the buildings of St Andrews Hall that remains; a detailed model was produced in 3Ds Max utilising both spline and box type modelling which are described in §4.4.1. As well as the spline and box type modelling, some tools were used to facilitate in the overall modelling process, this included the use of the "Wall modifier" tool in 3Ds Max, which allows the user to set the dimensions (thickness and hight) of a specific wall and 3Ds Max will fill in the corners automatically at any angle as shown in Fig4.8. Once the interior and exterior model of the cloisters were completed the textures were produced from photographs taken and edited in Adobe Photoshop. If a

Figure 4.6: Norwich Cathedral Cloister Exterior

Figure 4.7: Norwich Cathedral Cloister Interior

texture produced an undesirable tiling effect (see §4.4.2) then Luxology's ImageSynth plugin was used to automate the process of creating a seamless texture for use within the model. For added realism, bump mapping made by easy by using CrazyBump (see §4.4.2), this was used on the majority of the textures and helped in creating a 3D effect without adding to the polygon count of the actual model.

As documentation for St Andrews hall in the early 18th century is very rare, it was very difficult to know exactly what the interior of the cloisters would have been filled with. Speaking with Prof. Rawcliffe, it was understood that cloisters in those times were mainly used as grain stores and "storage rooms", there is however

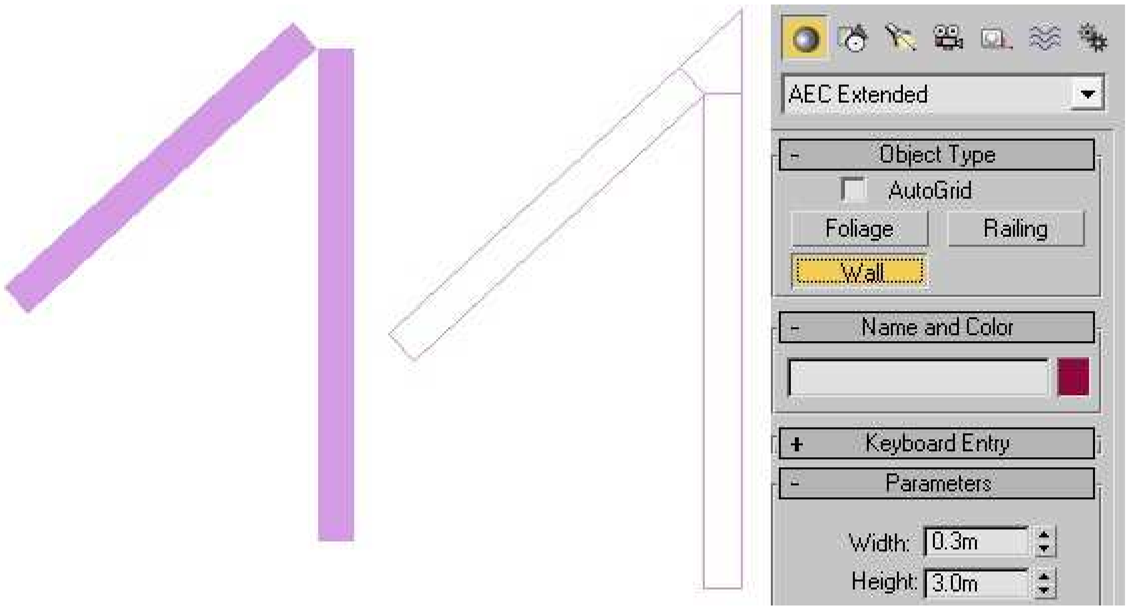

Figure 4.8: Wall Modifier Tool Example

documentation which shows that at some point there was a mint in one of the corners. Image references of grain sacks and weighing scales among other things for the time period were searched on Google. It was also known that a ledger was kept in the main grain store to keep track of peoples' accounts and debts.

Research showed that a library (Fig 4.9) had been placed in one of the upper floors of the cloisters, however this model could not be included in the final rendering, as it was later found that it had been completely ransacked in the early 17th century.

Figure 4.9: Authors' artistic impression of what the library may have looked like

4.4 Special Techniques used

This section will describe special techniques used during modelling and rendering to achieve an overall final effect.

File management

In order to work and manage files efficiently and effectively Xref (External Reference) files were used in the entire project. Xref scenes allow the user to load an external file temporarily into a current file, which includes all necessary lighting, texturing and geometry. Xrefed scenes cannot be altered in any way, but can still be edited in their original file and then refreshed in the target scene.

To avoid a slow down of computer memory and to increase overall productivity, all the high detailed internal models used to fill the scene were modelled separately in their own file, duplicated and then placed by Xref in the correct area of the scene.

Xrefs also provide the ability to be toggled on and off. This proved to be highly advantageous as different areas of a scene could be hidden in order to render out different areas at much faster speeds, this also included the ability to toggle between lighting setups for test renders to determine which had the best atmospheric effect.

4.4.1 Modelling Techniques

When modelling in 3D space, there are several alternative techniques that can be used to model the same object, this section will describe the difference between Box and Spline modelling as both techniques were utilised to complete the St Andrews Hall Cloister model.

Spline Modelling

The term spline is frequently referred to as a parametric curve. It is the simplicity of their construction, that makes them a popular choice when designing digital models as well as their interactive curve design that allows for easy manipulation [Art10]. Spline modelling was used extensively in the modelling of the window frames in the cloisters (Fig:4.10), as intricate detail was needed using smooth curves.

Figure 4.10: Example of Spline Modelling

Box Modelling

Box modelling is the technique used to model detailed shapes from the basic cube. The cube can be divided and subdivided into many sections, where its vertices and faces can be manipulated to transform its shape. The main shape of the cloister buildings were modelled using this method.

4.4.2 Some Special Texturing Techniques

In some cases, it is possible to save on geometry and achieve a similar level of detail using different texturing techniques. This section will give examples of some of these features.

Bump mapping

Bump mapping uses the intensity of the bitmap to raise or indent the surface of an object (See Fig: 4.11). The white areas of the map are raised, and darker areas are lowered. Although bump mapping appears to alter the geometry, in fact it does not. This gives the illusion of detail without actually having to increase the polygon count and render times.

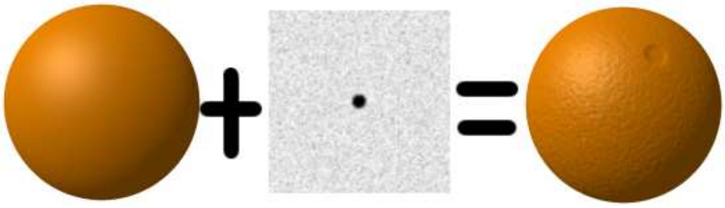

Figure 4.11: An example of Bump Mapping.

The simplest way in which Bump mapping is created, is to save a greyscale version of the texture and import it into the material editor in 3Ds Max. While this may produce good looking effects it does not always produce the best quality bump map. In order to achieve this, software such as CrazyBump [cra], can be downloaded.

Tiling

Tiling is the process of placing a copy of the applied bitmap on a selected object next to the current one until the entire surface is covered with the bitmap. In some cases, when the bitmap is repeated it ends up leaving a very prominent edge, and it is easy for the eye to spot where the bitmap starts and ends. To conceal this problem it is good practice to use seamless tiling, this is where the texture repeats from edge to edge and where the horizontal and vertical seams line up. This allows for a bitmap to be duplicated with a more natural look and feel.

Tiling was used with many textures other than the stone wall shown in Fig. 4.12, such as the wooden flooring, and interior detailing. To achieve a seamless texture, it is possible to manipulate the image in Adobe Photoshop or any drawing package, or with specially designed image manipulation software such as The Seamless Texture

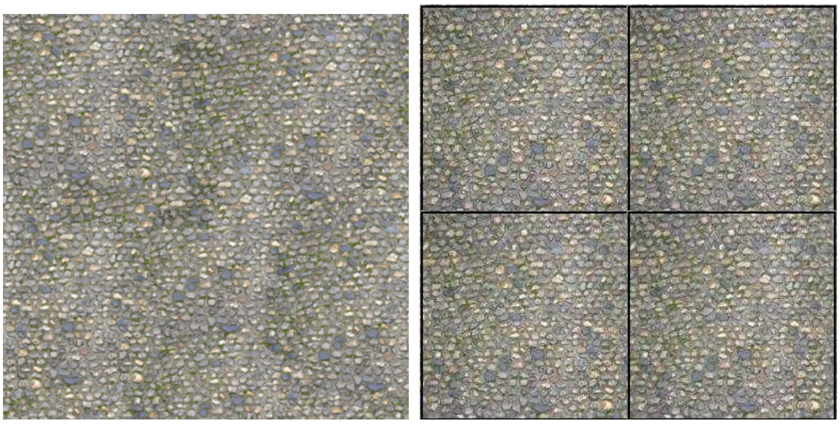

Figure 4.12: An example of a Seamlessly Tiled and non-Seamless tiled texture

Generator and Luxology's ImageSynth.

Opacity Mapping

In some cases, it is necessary to have transparency in certain textures. Opacity mapping determines which areas are visible and which are transparent. Black areas for this map are transparent, and white areas are opaque. Fig. 4.13 shows an example of how Opacity mapping is created for the window.

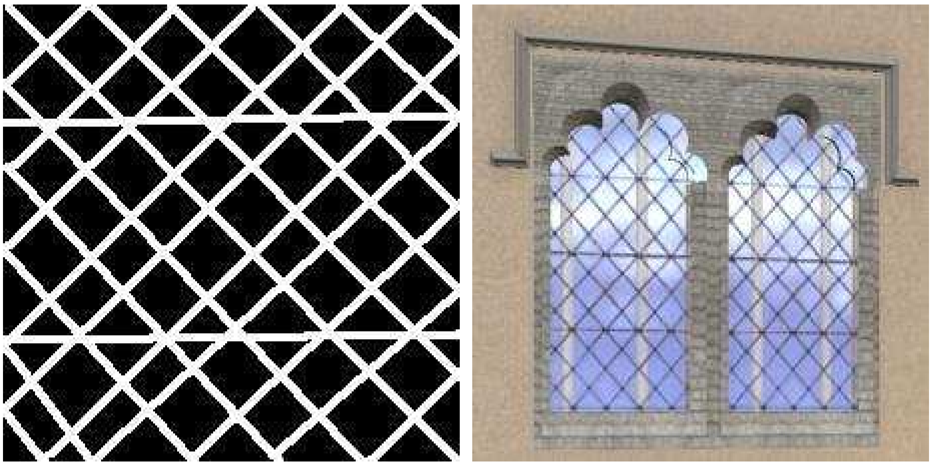

Figure 4.13: Opacity map to produce final glass texture

Chapter 5

Interfaces

There have been many theories regarding human computer interfaces, such as looking at the psychology of human interaction with a graphical user interface, to the size and placement of buttons and links on websites. This chapter will describe the best practices for user interfaces.

5.1 Laws of Human Computer Interfaces

Human Computer Interfaces are key to the success of any end user software. That is why it is vital for program designers to get this right. The target end user needs to remain the primary focus throughout the design and development phase of the application. The following sections will discuss current uses of interface designs in general and then look more closely at HCI for heritage applications.

5.1.1 Hick-Hyman Law and Fitts' Law

Hick-Hyman Law and Fitts' Law are two surviving performance principles based on Shannon and Weaver's [SW49] Information Theory. In the early 1980s, these laws were presented as design principles for developers to maximize usability in the design of human-computer interfaces. This next section will describe both Laws, and talk about how they can and may be adapted for use within this project, and why Fitts'

Law is superior when it comes down to developing HCI. Hick-Hyman Law, also known as Hick's law, describes the time that it takes a person to make a decision based on the choices that he/she has been given.

Hick - Original Experiments (1952)

Hick [Hic52] devised an experiment that contained 10 pea lamps arranged in an irregular circle formation connected to a device that was punch-tape coded to light one random lamp every 5 seconds. Each lamp, had its own corresponding "Morse key" that was placed on all 10 fingers of the participant. The task was to depress the correct key to light a particular lamp. Both light and response were then recorded in binary code by moving paper.

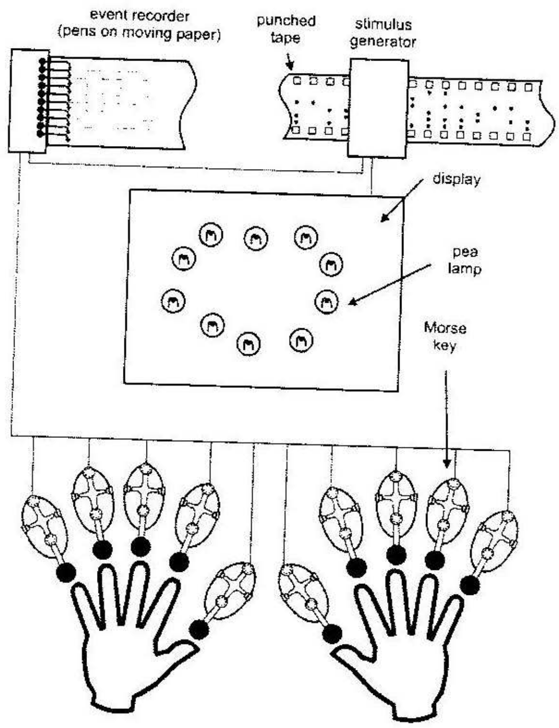

Figure 5.1: Authors impression of the apparatus in Hick (1952) [Seo05a]

Hyman - Original Experiments (1953)

Hyman [Hym53] expanded on Hick's experiment by altering the probabilities of the stimuli (i.e. in this case, the lights) such that they are not equi-probable to yield varying results so that he could assess the reaction time, RT, as a function of the bits per stimulus H_T . Hyman used 8 lights in a matrix of 36 lights, and designated names to each of them: Light1 - Bun, Light2 - Boo, Light3 - Bee, Light4 - Bore, Light5 - By, Light6 - Bix... and so on.

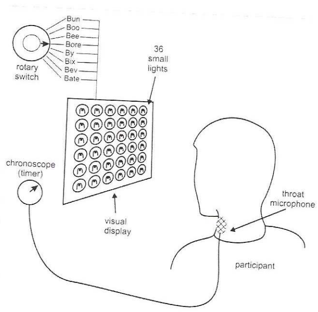

Figure 5.2: Authors impression of the apparatus in Hyman (1953) [Seo05a]

At the beginning of each trial a warning light would turn, then one of the eight lights would flash for two seconds while a timer started simultaneously. Participants would react to the light by calling out the particular name of each light. In this case a throat microphone was attached to the participant and activated an electronic voice key to stop the timer. Hyman found that RT was linear as a function of bits of the alternatives.

Therefore with the extension of Hyman (1953), Hick's Law was consequently accepted by many as the Hick-Hyman Law, which predicts a linear relationship between reaction time and transmitted information: $RT = a + bH_T$ (a and b are empirically determined constants, and H_T is the transmitted information).

5.1.2 Fitts' Law

Fitts' Law is the other surviving Information Theory model in psychology. The law states a linear relationship between task difficulty and movement time.

Fitts' Original Experiments

Fitts conducted 3 experiments [Fit54]. Experiment 1: participants used metal-tipped styli, varying in weights, 1oz was used on the first day of the experiment and 1lb used on the second day of the experiment. The task was to tap two stationary strips of metallic targets. These target widths (W) varied from 0.25 to 2 inches, and the distance between them (A) varied from 2 to 16 inches. The participants were asked to strike the target places alternately to score as many points as possible, therefore accuracy was encouraged.

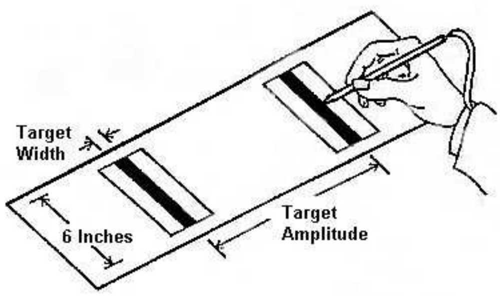

Figure 5.3: The serial tapping task used by Fitts (1954)

Experiment 2: Participants were required to move and stack round plastic disks (with holes drilled in the middle of various sizes) from one pin to another with various diameters. Experiment 3: similar to experiment 2, however this time participants were required to transfer the pins from one set of holes to another. There are no elaborate results for experiments 2 and 3, however for experiment 1, Fitts found the most difficult condition occurred when the smallest W and the largest A, yielded an error rate of 3.6 % with the lighter stylus and 4.1 % with the heavier stylus.

5.1.3 Fitts' Law on HCI applications

The principle of human movement was published in 1954. The law states a linear relationship between task difficulty and movement time from a starting block to a target area (§5.1.2). The way in which this is related to websites describes how at any given time on a website there are five main points that can be easily selected. Fig.5.4 shows the points on the screen assuming the user is right handed. The first and easiest place to click is where the mouse is currently positioned on the screen at any given time, the other four places are each of the corners.

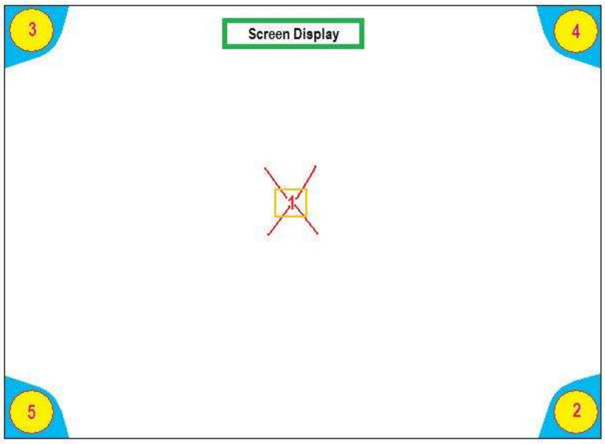

Figure 5.4: Example related to Fitts Law

MacKenzie [Mac92] summarised six studies that applied Fitts' Law to HCI. Among them is Card, English and Burr [SC78], regarded as the first application of Fitts' Law in HCI. There are many works applying Fitts' Law to the use of pointing devices, however only selected topics will be mentioned here.

Pointing: Fitts' Law has been applied to evaluate many pointing devices. The use of a mouse, a trackball, and a tablet with stylus were investigated by Kabbash, MacKenzie, and Buxton [KMB93] with the application of pointing and dragging experiments.

Text Entry of Soft Keyboards: Fitts' Law has also been very successful with the ubiquity of hand held devices, such as PDAs and cellphones, and the increase in their functionality and text entry on their GUI [Seo05b]. A more recent development is the tapping of a stylus on a soft keyboard, or a graphic representation of a computer keyboard. An evaluation into six different keyboard layouts - QWERTY, ABC, Dvorak, Fitaly, JustType and telephone have been conducted by Mackenzie, Zhang and Soukoreff [MZS99], who reported the typing speeds of novice and expert users. Out of all layout styles the QWERTY keyboard was immediately seen as the superior layout, as users were able to produce faster and more accurate typing speeds.

Navigation: Another area that has seen the application of Fitts' Law is controlling navigation within a GUI environment such as panning and zooming. One such application is multi-scale pointing, which extends Fitts' paradigm beyond 1D (one dimension), by allowing users to zoom and pan [AJS97].

5.1.4 Banana Principle

The Banana Principle is based on the idea by Seth Godin [God02] which states that if one was trying to guide a monkey through a maze it would be compelled to follow a route with bananas (these would be the monkeys "targets"). In the same way for a mouse, one would offer cheese. Therefore the idea that Seth Godin is trying to suggest the user be given something to urge them in the right direction.

Comparing a few Websites

The idea is that on a web page, the author must create a banana so that the user will follow the lead and click on that link or banana. This section will compare two search engine sites, Yahoo and Google, and two Technology sites, HP and Apple.

Looking at the Yahoo homepage it is very overcrowded with a lot of information on one page, many different links from shopping to instant messaging as well as advertising. This creates hundred of different entry points into the site from the home page. It seems that Yahoo are trying to cater for all people, by providing as much information as possible on a single page. However there is not one clear link entry point. By applying Godins' principle, Yahoo should try and eliminate most links and information on that page. Comparing the Yahoo homepage to that of Google (Fig 5.5), shows the difference where Google's contains a huge Banana on the front page, that being the search box which appears just under the logo. It is simple and clean which makes it easier for the user to find.

Figure 5.5: Screenshot of the Yahoo and Google Homepages.

While looking at another two websites in the same industry (Fig 5.6), the HP UK and Apple UK websites, HP has many links in different boxes, but the user still has to hover over the exact words to click on each link rather than the box itself, therefore not applying Fitts' law correctly. Although the web page does have a large banana for users to click, situated on the right hand side of the screen. Even though

Apple have over 30 entry points into the website, these are located at the bottom of the page and the majority of the website is dedicated to a large banner in the centre of the screen, in this case Apple are promoting their latest operating system, thus giving the user a big target to click on.

Figure 5.6: Screenshot of HP and Apple Homepage

Therefore eliminating most design elements on a page, and focusing on a single message, is good UI practice, allowing for the most important item to have the most important space.

5.2 Different Interaction Techniques

This current section will look at various interaction techniques between humans and computers.

5.2.1 Hardware interaction

Many ways exist in which a human can interact with a computer using different hardware technologies, these could range from a basic 2D keyboard and mouse, to a

Figure 5.7: SpaceNavigator by 3Dconnexion [3Dc]

3D controlled peripheral such as a Joystick, rollerball, or by using the newly marketed SpaceNavigator ©by 3Dconnexion which allows the user to pan, zoom and rotate, as if they were holding a model in their hands [3Dc].

There are hundreds of input devices currently on the market, each with its own set of objectives and uses. This section will talk about relevant input devices and possibilities of expansion to the project. Our main concentration will be aimed at 2D devices, as this is the standard hardware a typical user will have, however a section on 3D devices, including haptic devices will also be discussed. Before continuing it is important to distinguish between input devices and interaction techniques. Input devices are the tools used to implement interaction techniques. Many interaction techniques can be mapped onto a single input device, whilst maintaining the ability for the device to naturally and efficiently work with the given technique [GHvD94].

CAVE

The CAVE (CAVE Automatic Virtual Environment) [§3.1.6] is a projection-based virtual reality system. Therefore large fixed screens are used to provide a panoramic display without encumbering the user. The CAVE is a 10foot-cubed room, where the user wears liquid crystal shutter glasses to resolve the stereoscopic imagery. An electromagnetic tracking sensor attached to the glasses allows the CAVE system to determine the location and orientation of the user's head [PAC+01]. The idea behind this is that the user can physically walk around an object that appears to exist in 3D in the middle of the CAVE. The user can also hold a "wand" which is also tracked and has a joystick and three buttons for interaction within the virtual environment. VR applications displayed on the CAVE system can be linked over high-speed networks. Therefore users can share the same virtual world from remote locations around the globe. They can interact with each other and with the objects in the virtual world and can also see each other's movements as avatars.

Input Devices

Traditional WIMP interfaces are still prevalent, but new technologies have allowed non-traditional devices and interfaces to become acceptable. These include devices such as 3D pointing devices, Head mounted Displays (HMDs) and haptic devices. Input devices can be split into three main categories based on the type of event they generate. Devices that generate one event at a time based on the user, are known as discrete input devices. Whereas continuous input devices generate a stream of events. The devices that can perform a combination of event types are known as hybrid devices. Examples of these are pen-based tablets which are becoming more popular in VR applications because they give the user the ability to interact in 2D [GHvD94].

POINTING DEVICES:

A mouse is a typical pointing device. However there are a few alternatives that have been developed such as touch screens, where a user touches a point on a screen to navigate to, this is usually achieved by detecting movement across a physical surface. Other devices such as joysticks, pointing sticks or 3D mice, work by reporting the angle of deflection. Movements of the device are then shown on a screen by movements of a cursor, this creates a simple way to navigate through a computers GUI.

DEVICES WITH HIGH DEGREES OF FREEDOM:

High DOF devices, include things such as fibre optic sensors, wired gloves which a user wears, and haptic devices. Haptic devices are now slowly growing in popularity and are becoming more intertwined with virtual reality [LLD06]. This adds a sense of touch as to previously visual only solutions. Newer Haptic devices such as the The Virtuose 3D15-25 (Fig5.8) work with 6DOF, this allows them to feel very natural when it comes to "feeling" around a virtual object.

Figure 5.8: The Virtuouse 3D15-25 Device

EYE TRACKING (GAZE) AND GESTURE CONTROL:

Vision based HCI include both eye-gaze pointing and gesture recognition [FMS02] which allows the user to "point" and travel by looking in the direction of an object or location on the screen. As well as providing greater intuitive interaction in a virtual environment, eye gazing also includes gaze typing systems for the physically challenged, such as the Eye-Switch system [KFW+79]. Today, portable head-mounted eye trackers are available (see Fig. 5.9) and the user is no longer required to remain stable. Eye tracking therefore has also become an attractive input method for augmented and virtual reality [PLW08]. As for gesture control, Project Natal [Res10] is a revolutionary new way to interact with virtual environments as it allows the user to become the controller in its gaming environment, without the need for a game pad or controller, however (at time of writing) this technology is very new and is in the final stages of development.

Figure 5.9: Portable head-mounted eye tracker [PLW08]

COMPOSITE DEVICES:

These are considered to have more than two different forms of input. With plenty of open source Application Programming Interfaces (APIs), and gaming manufacturers of hardware increasingly releasing APIs for their products, it has now become easier to program composite devices to be used as user input. Many gaming devices contain composite controllers, such as the remote for the Nintendo Wii. In certain environments the use of a Wii input device, would be considered far more exciting and feel more interactive than a standard mouse input.

5.2.2 GUI interactions

This project will concentrate mainly on the use of GUIs for the application, as the software being made is eventually to be used online for people to view from their personal home computers, however an expansion on this could be at exhibitions and museums where different input devices would need to be considered. This next section will discuss the pros and cons of each of these devices that could be relevant in the use of this project. GUIs allow a user to interact with electronic devices, in the case of this project - computers. The user interface design of a VR system is a critical component needed in any user friendly application. User interfaces are now becoming far more diverse, and people often find it inherently difficult to understand 3D spaces within a 2D environment (i.e. Computer screen). GUIs combine the uses of different technologies and devices to provide a platform for users to interact with. The most common combination in GUIs is the WIMP paradigm, especially in personal computers. Therefore it becomes clear that simply adapting traditional 2D WIMP interaction styles to 3D systems does not solve the problem, rather, new 3D user interfaces, based on real-world interactions must be developed with haptic and tactile displays being present. Interaction designers must have a thorough understanding of the ergonomics, advantages and limitations of the devices used so that intuitive mapping techniques can be found between interaction techniques and hardware [BKL+00]. A common error of 3D user interface design is that 3D applications should only utilise 3D interaction. In reality, it is seen that 2D interaction offers a number of advantages in certain tasks, such as efficient selection techniques as interaction on a physical surface, which provides the user with a sense of feedback (such as writing and annotating). Therefore when using both 2D and 3D interaction techniques, developers create interfaces that are easier and more intuitive to use.

Interaction Techniques

Navigation is the most prevalent user action in 3D environments, therefore it is very important that navigation is made easy so that users can focus on more important tasks, such as learning (understanding and reading facts about the area) and visual stimulation, (the ability to look around the 3D scene). There are five main categories [GHvD94] in which "travel" interaction techniques fit, this section will only discuss the two types that are relevant to this thesis:

- *Target-based travel:* the user specifies the destination and the system handles the actual movement. This may take the form of "teleportation" where the user is taken from one area to another, or, the system may produce some sort of transitional movement between the two locations. This technique is very simple from the user's point of view.

- *Route planning:* the system follows the path and controls the actual movement that the user pre-defines through the environment. These techniques allow the user to control the travel while retaining the ability to do other tasks during the motion.

The system developed for this project, will rely on Target-based travel techniques as pre-rendered animations of the 3D environment will be used. Therefore as seen in this chapter, there are a number of points that are needed in order to achieve a good quality HCI:

- Restrict/Limit the required DOF for input wherever possible and provide constraints to help guide the user and avoid confusion with many inputs and buttons on screen.

- If taking advantage of novel input devices, give the user structure and support in managing the increased complexity.

5.2.3 First Public HCI for Historical Urban Modelling

According to the Dudley Castle online archive [Joh08], it was the first concept of a virtual tour, that was opened by the Queen in 1993. The Virtual Tours system is the work of Colin Johnson. It is a system designed to be operated by anyone with no prior knowledge, skill or dexterity. It is intended for museum and exhibition environments, where a virtual helmet type application is not recommended. The system works by allowing the user to be guided along pre-set paths by the voice of an historically liked character. Options are given at certain junctions to turn right, left or straight on, thus giving the user control over direction without allowing them to lose control and get lost within walls or floors etc. The system itself uses images displayed at video resolution (768x576) to a large screen projection system, all at a full frame rate of 25 frames per second (FPS). It was described that the system will have to be used by many people, including individuals and groups, therefore everyone had to be catered for, and for that reason a fully immersive VR headset was ruled out, due to their "insular" experience, hygiene and restriction of visitor throughput. At the time, machines were not powerful enough to support high poly real-time rendering. Therefore it had to be ruled out, because the image quality required for producing good looking shadows and reflections at video resolution meant that any machine which could produce that at 25FPS would be so expensive that it would prove to be an unreliable and costly option.

5.3 Evaluating Interfaces

As described in this chapter, interfaces can be very intuitive and difficult to achieve. Therefore it is very important to take the laws of HCI into consideration when creating one. The point being, to allow users of any computer ability, the chance to quickly and easily use and navigate around a system with minimal or no help required. As technologies improve, people will continue to develop new ways in which to interact with computers and computer systems, as seen in §5.2.3 from the first public HCI used by the queen - to current 6DOF devices and haptic feedback as described in §5.2.1.

Chapter 6

3D Online

6.1 Video Encoding

There are few encoders on the market for video playback in user browser windows, the current two dominant plugins include the Flash plugin and the QuickTime (QT) plugin. In this section we try to ascertain "which one is better?". To respond to this, the author compared the pros and cons of each plugin, as well as looking at market share/penetration and global usage of each.

It is useful to look at major global video sites such as Youtube and Vimeo and determine what they use. With the large market penetration that Flash has, there is a high probability (0.99 [see Fig: 6.1]) that the user already has the plugin installed on their computer. This means that the user will not have to navigate away from the website to download the player.

Flash is a non-biased player, this means that it was developed to be used across all platforms, unlike QT which was created mainly for Apple Macs. It's because of this that many PC users choose not to download that player.

The difference in video quality between the two players is minimal, this is even more so now that Flash has enabled H.264 video compression. H.264 provides great video quality and does so across the bandwidth spectrum from 3G (mobile phones) to HD (High Definition), while using up about half the bandwidth of MPEG-2 (another

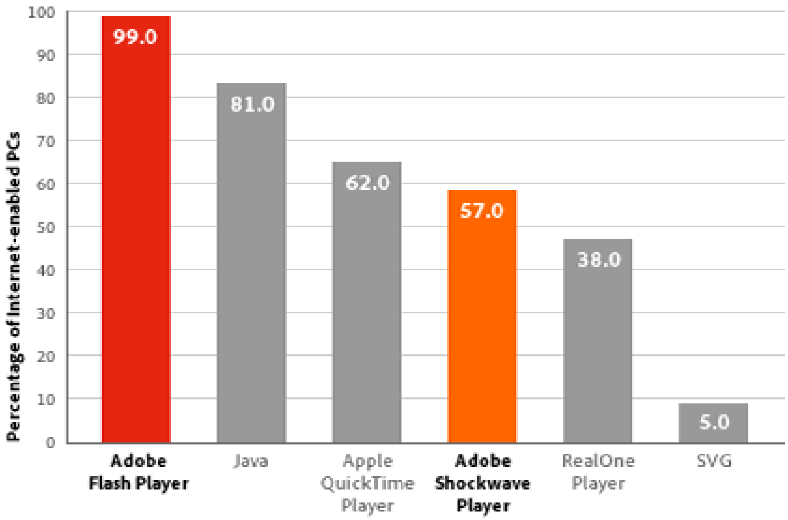

Figure 6.1: Millward Brown survey, conducted June 2009. (Markets include US, Canada, UK, France, Germany, Japan).

compression format). This is why H.264 is very popular and is used to broadcast HD television by BBC HD, SKY HD and many other companies.

		Flash	QuickTime (Free Version)
Customisable Skin	Development Environment	Yes (Free)	No (only Pro)
	Viewer	Yes	Yes
Market Penetration		99%	62%
Full Screen Mode	Development Environment	Yes (Free)	No (only Pro)
	Viewer	Yes	Yes
H.264 Compatible		Yes	Yes

Table 6.1: Table showing comparisons between Flash and Quick Time

6.2 Flash Player

As section §6.1 showed, Flash is more widespread, and reaches a greater audience than any other standard, it is now also pre-loaded in most browsers, so users do not even need to install it in some cases. As the author is looking to use technology that is easily understood and used by the majority of users, such as those who are not computer literate, it becomes the clear option in choosing to encode videos in Flash instead of any other form. As well as the ability to reach a very wide audience, Flash has had many plugins developed for it, which makes it highly customisable using its native scripting language Actionscript (AS). The player itself, as well as being free to download also offers customisable skins to allow it to blend into the design of any website or system as seen in Table 6.1. Flash itself, simply refers to both the multimedia authoring program, and the Flash Player. Technically, Flash is the authoring environment and Flash Player is the virtual machine that is used to run the Flash files. Originally the Flash player was designed to display 2D vector graphics and animation, but it has now moved onto 3D object transformations in Flash player 10. Due to the fact that vector graphics are used, allows the file sizes to be as small as possible, hence saving bandwidth usage and loading times.

6.3 Importing 3D models into a Flash environment

In the past there have been many ways of "faking" the way in which 3D models are used in Flash. The most commonly used technique consists of taking many images of the object from different angles, and applying some sort of skimmer GUI that shows each image in sequence, as if the 3D object itself was being displayed. Many current mobile phone websites use this method. Currently there are few programming languages that help produce 3D models for the web, such as Swift3D, VRML, Web3d (X3D) and Flash Actionscript. This section will look at each individual method, and discuss various techniques and technologies associated with each language.

6.3.1 Swift3D

ElectricRain's Swift3D application allows the developer to create 3D animated content in an application such as 3DS Max, and then use the Swift3D software to convert the content into the Flash SWF format.

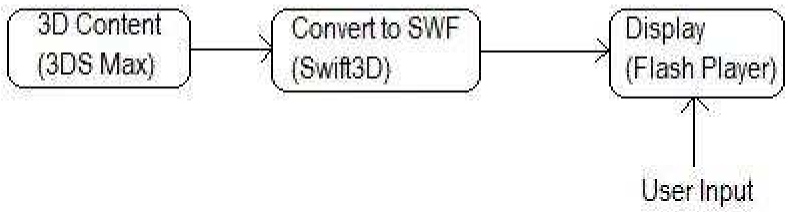

Figure 6.2: Swift3D Workflow

Figure 6.2 essentially shows how an animation can be created in an appropriate application, by using Swift3D to convert the animation into a sequence of 2D bitmaps or vector graphics, the third application - in this case Adobe Flash can be used to display the results in the browser window [McC04]. The only draw back to this workflow is that it effectively produces a non-interactive form of 3D graphics, as the player can only access the 2D version of the 3D content, as well as it being an off the shelf program and therefore not free to download.

6.3.2 Freespin3D

Freespin3D is another Flash extension and also like Swift3D, it is not free. However it does contain an extensive API for developers. It is the only real-time 3D technology that utilizes the standard Adobe Flash animation time line and built-in 3D behaviours, making FreeSpin3D the most intuitive solution for interactive designers [fre09]. As mentioned in §6.3 many techniques were used to artificially create a 3D effect of objects in real time, whereas Freespin3D allows for 3D models to be directly imported into Flash environments, as well as interactive or animated content.

6.3.3 VRML & Web3D (x3D)

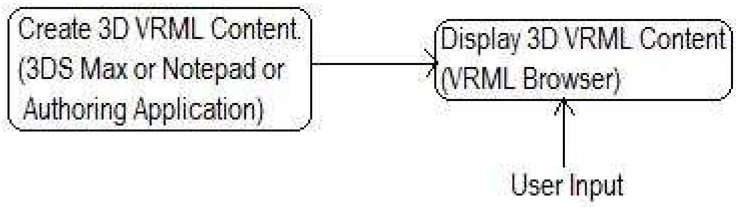

Figure 6.3: VRML Workflow

VRML (Virtual Reality Modelling Language) was the original programming language that allowed 3D interactive vector graphics to be displayed specifically for the web [Gra03]. It was the first standard of its kind to be agreed by the Web3d Consortium, which was created to further the development of the format. While the original VRML program was designed in 1994, it was later upgraded in 1997 (commonly known as VRML97). Although it had a strong programming language for user interactivity, it failed in the areas of stunning visual graphics. In its time it was unable to cope with high poly*(gonal)* models, which are necessary for a photo-realistic environment. In order to display any type of 3D content VRML would need to follow the workflow presented in figure 6.3.

X3D

X3D is a royalty-free open standards file format and run-time architecture to represent and communicate 3D scenes and objects using XML [Con08]. It builds on the VRML system by adding different extensions. Designed to support 3D graphics and programmable shaders for higher visual quality graphics, as well as animations and mouse based user interaction (such as clicking and dragging). Its new modular architecture allows for layered profiles that can provide increased functionality for immersive environments and enhanced interactivity.

Papervision3D

Papervision3D (PV3D) is an open source 3D rendering engine for Adobe Flash that allows real 3D models to be imported and manipulated in Flash using Actionscript. PV3D relies on industry standard XML schema, which is used to describe a wide range of 3D features. These files are known as COLLADA (COLLAborative Design Activity). These files are recognised by their file extension ".dae". COLLADA was originally developed by Sony Computer Entertainment in order to create a standardised way of exchanging digital assets. For the purpose of PV3D, COLLADA files contain data from 3D models. These files are exported from a 3D program and reinterpreted by the PV3D engine.

6.3.4 Actionscript 3

There have been many attempts at providing a mechanism that would allow the delivery of interactive 3D content on the World-Wide Web, as seen previously in §6.3.3. However the majority of these solutions have required proprietary plugins on the user's browser and so far, none of these have become a browser standard [McC04]. On the other hand, Adobe Flash, has nearly achieved the status as the standard plugin for displaying interactive multimedia and graphics.

6.4 Comparison of 3D technologies

Table 6.2 will attempt to show the differences and comparisons between all the software mentioned in this chapter. Please note that each program/extension has many more features than mentioned in the table, those chosen are features that could be related to this project.

Since writing this chapter, Google Code labs released Google O3D on April 2009. Although O3D is not Flash based, the author decided it was necessary to address the

	Swift3D 6.0	**Freespin3D for CS4**	**VRML 2.0**	**X3D**	**PV3D GreatWhite**
Open-Source/Free	No	No	Yes	Yes	Yes
Ability to Export 3D models into Flash	Yes	Yes	Yes	Yes	Yes (needs additional free plugin for Max (Collada))
Special Features	Faster rendering speeds/styles, two pass curve fitting (attempt to fit together groups of edges with a curve)	Animate using the Adobe Flash timeline and motion tween, edit 3D Model Textures on the fly and apply animated textures	Original standard for online 3D in real-time (has now been discontinued)	Real-time graphics are high quality, interactive, and include audio and video as well as 3D data.	Various rendering engines as well as linear texture mapping, optimized for rendering speed and quality
Ease of Use and Installation	3D Max Plugin easy to install	Adobe Flash extension easy to install	Discontinued (however special browser was needed to play VRML applications)	Requires freely downloadable VRML/X3D editor/browser such as FLUX or freeWRL	Repositories downloaded through an SVN client, then had to be linked in with Adobe Flash

Table 6.2: Table showing comparisons of alternative Flash plugins

latest technology for delivering real-time 3D content on the world wide web through the basics of popular web browsers. This allowed for rendering inside the Browser Window, it is open source as well as cross platform and browser compatible (Firefox, Safari, Google Chrome, and Internet Explorer). However, it is still in early development, all that is required is the O3D plugin to be installed in the browser. There is a moderate API and developers guide included online. O3D extends Javascript methods by using standard JavaScript event processing and callback methods, as well as having the ability to import files from the COLLADA extension format [O3D09].

6.5 Conclusion

Flash has evolved throughout the growth of the Internet and for the foreseeable future, will continue to be a part of web interaction and Internet media. It has cemented its authority as the best browser plugin for movies and animation playback. As documented in Chapter **??**, Papervision 3D is fairly simple to use, as well as being open source it is free for all to download and modify, has many extensions that can be added for feature expandability, as it works well with Flash, it is the most logical choice due to the huge demographic reach of the Flash component. However due to it being relatively new technology, a high learning curve is attributed to the fact that there is little official documentation. [note since writing, the first PV3D book has been published (September 2009) and is printed on demand; as of yet it can only be purchased online].

Chapter 7

Program application

7.1 Design approaches

Based on the research conducted in chapter 3.1 on current or previous technologies used, the author began by sketching basic layouts for the GUI design based on ideas used to display similar elements. The evolution of the design can be seen in §**??** of the Appendix. A screenshot of the final design is shown in Fig.7.1. The arrangement has been laid out in such a way that it is easy to use, with the main virtual walkthrough dominating the centre of the screen,

7.2 Papervision3D

Papervision3D (PV3D) is a 3D rendering engine that allows real 3D models to be imported and manipulated in Flash using Actionscript. At the time of writing PV3D is still in beta testing, and does not yet have any official book. Most resources used were in the form of online tutorials as well as trial and error methods.

7.2.1 Installing PV3D

PV3D is not a stand alone software application, it is a set of add-on extension classes for Flash Actionscript 3. The repositories had to be downloaded through a SVN (Subversion) client - Tortoise SVN was chosen as it was free to download and use.

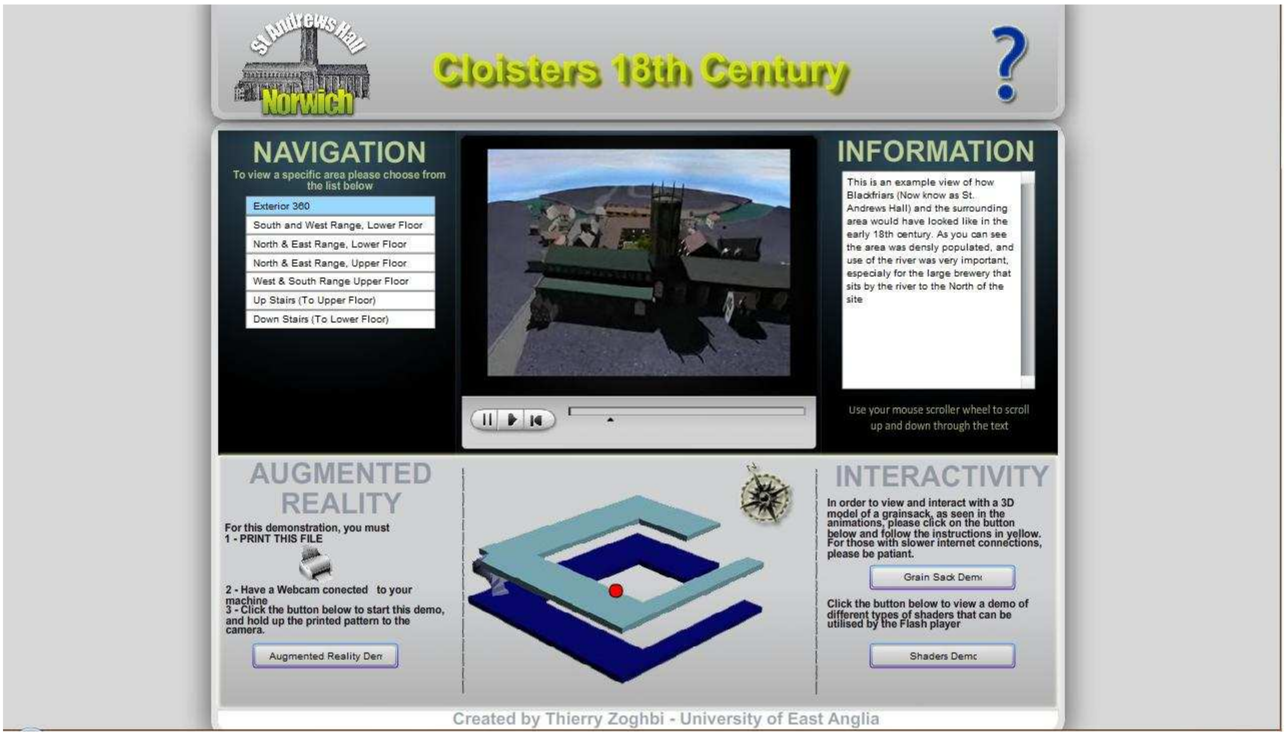

Figure 7.1: Example Application for Virtual Walkthrough Screen shot

Once Tortoise SVN was installed on the computer system, it was necessary to direct it to the following web address http://papervision3d.googlecode.com/svn/trunk where the PV3D classes were stored, and download them directly into a folder onto the authors computer system. §7.2.2 will explain how to export 3D models from 3D Studio Max and import them into Flash.

7.2.2 Exporting/Importing 3D models

Before exporting can be done, the COLLADA plugin [col08], was needed to be installed in order to use the ".DAE" file type which PV3D uses to read details of the 3D model, it is XML based and can be opened in notepad to view properties of the object such as the material name.

Preparing the 3D model for Export:

It was found that best results were obtained when the model was kept to a low poly design (preferably under 20,000 polygons - see §7.5) and if the model was made of many parts, it should be all attached and exported as one whole object. If texturing was to be added, it should be applied using the UVW unwrap method and applying the material using the "diffuse channel". If lighting and shadows are needed, then they must be baked onto the texture itself. In order to do this, lights must be included in the scene, and the rendering dialogue tab "render to texture" should be used, which will save the lighting effect onto the texture itself. This will give the illusion of real light in a scene without the need for it (however, if the object is to be animated in Flash, this option is not recommended as the lighting will not change as the object is moved, and this will look incorrect). When everything is completed, the user only has to navigate to:

File $\rightarrow ExportSelected \rightarrow .DAE$

In the Flash Drop down menu and save the file in a folder which can be used at a later stage with Flash AS3.

7.2.3 Rendering Engines

Every Papervision file requires four parts in order to work correctly, these include a Viewport, a Scene, a Camera and a Render Engine. There are many types of Rendering Engine in Papervision, including Lazy Render Engine, Quadrant Render Engine and Basic Render Engine, each of these engines specialise in displaying objects differently, however the author will concentrate on the Basic Render Engine as it has the basics ready for most types of scenes.

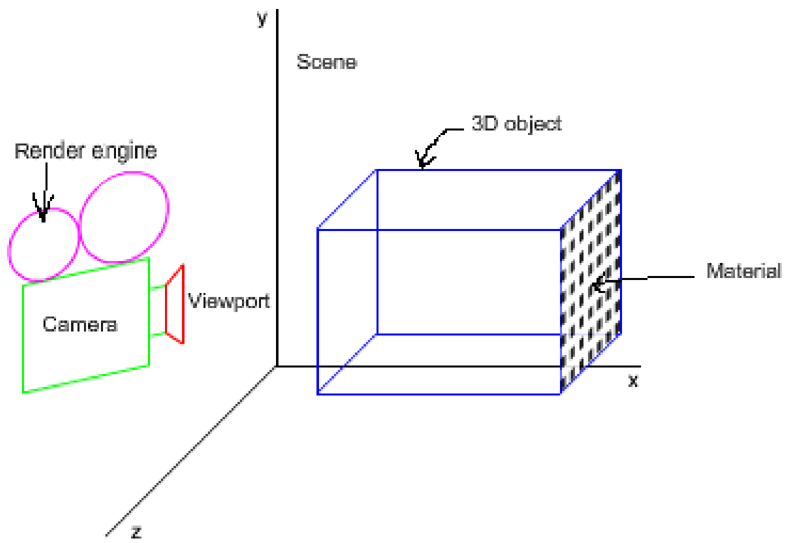

Figure 7.2: Basics of a 3D scene

Viewport: The viewport displays the rendered scene. Its a sprite that displays all of the objects contained in the view frustum.

Scene: The scene is where objects are placed. It contains the 3D environment and manages all objects rendered in Papervision3D.

Camera: The camera is the user's point of view, as if they were inside the 3D scene and defines the view from which a scene will be rendered. When rendering, the scene is drawn as if the user were looking through the camera lens

Render Engine: The rendering engine draws a sequence of frames and displays them in the viewport in order for the user to see an animation, this sequence of frames is kept in a loop, using a timer or onEnterFrame function in Actionscript.

The Basic Render Engine

To begin writing code for a 3D scene, certain variables have to be added, this can be easily written in Actionscript as shown below:

```
public class Main extend Sprite{
        private var viewport:Viewport3D;
        private var scene:Scene3D;
        private var camera:Camera3D;
        private var renderer:BasicRenderingEngine;
```

However to save time, and for cleaner code, BasicView can be used which enables the Basic Render Engine that contains all of the necessary variables as shown above ready for immediate programming of the functions.

```
public class Main extends BasicView

public function DrawSphere()
{
...
}
```

This method is most commonly used among users of PV3D. However there are many instances where other more complex rendering engines need to be imported, usually this allows for programmers to override individual variables needed for the 3D scene, such as specifying the size of the viewport to predefined values, rather than allowing BasicView to scale the object to the size of the stage by default.

7.2.4 Adding Materials & Shaders

Figure 7.3: 6 Shader Examples

Fig 7.3 shows some of the many different shaders and materials that can be added to objects using PV3D, these include: Composite Shader Material; Colour Material Shader; Flat Shader Material; Environment Map Shader Material; Phong Shader Material. Each of these shaders and materials vary in complexity of processor usage, some require an additional rendered light source, and some require external reference images. Shaders are a way of adding colour and "texture" to an object in real time. This basically works by taking the 3D object, the camera and some light source, and outputting a coloured pixel on the screen. This section will describe the methods used to achieve some of the more complex shaders.

Environment Map Shader

To create a glossy, reflective surface, the Environment map shader can be used to reflect a background texture onto an object. The method requires a texture imported from the background and uses it as a source for creating a reflective material, giving the object a realistic reflective coating which has been "faked" due to the fact that the graphics engine is not rendering a reflective shader, but using a pre-drawn texture. It keeps processing requirements down, while giving a nice effect.

Initially it is necessary to import the .dae file as described in §7.2.2 by writing a set of functions that will also add the environment map texture and the light source. Once this is done it is necessary to add some type of animation to the light source, or to the object itself, in order to make full use of the visual effects.

The reason that the Environment map shader is used for this type of effect, is that Ray-Tracing is a slow process. The calculations involved with lighting, reflection, and shadows are done individually on each pixel, and that taxes the CPU tremendously.

Phong Shader Material

The Phong Material accepts 4 parameters, the object, the light colour, the ambient light and the specularity (glossiness) of the material. The Phong Material displays as though light is shining through the object (almost as though the light source is inside the object, giving the object a slightly translucent effect). Light hits the object on both sides and based on the value of the specularity (0-100) the areas of light intensity vary in radius size and blend off to the ambient light colour.

7.2.5 Augmented Reality using FLARToolkit

In order to encompass a large variety of HCI, the author looked briefly into the use of augmented reality. For Papervision3D and AS3, an open source class library extension is the FLARToolkit [fla09]. It works by recognising a marker from an input image using a camera and then calculating its orientation and position in a 3D world.

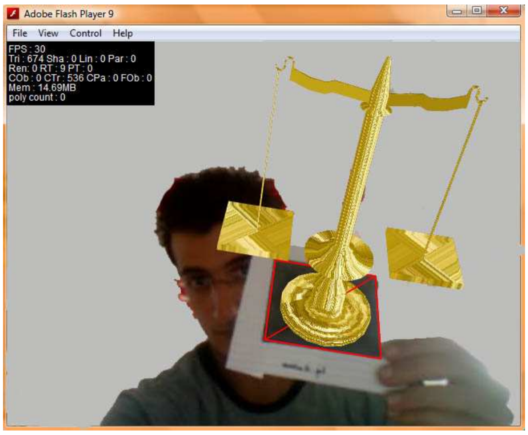

Figure 7.4: Augmented Reality using FLARToolkit

To produce the .pat file type needed for FLARToolkit to read and understand the marker being displayed, it is necessary to go to the Marker Generator Online [Mar] and use it in order to output the necessary file type. Once this is done, Actionscript will load a 3D model associated with the marker, and the user will be able to rotate and view the object in 3D as seen in Fig7.4.

As described in §3.2.2 it is human nature to hold objects in ones hands, this method brings much of this inquisitiveness to reality. For instance, if the user wanted to look in more detail at a specific artefact or object such as a statue, then all the user would need to do is pick up a marker with a picture of a statue on it and hold it in front of the camera to view it in 3D. The user will then be able to move it around in which direction they choose.

7.3 Semi-Real Time Virtual Walkthrough

This section will describe how the virtual walkthrough was designed in order to give the sense of real-time navigation, when in fact all animations were pre-rendered.

7.3.1 How was it designed?

Each area of the entire model was split into individual sectors, and rendered out separately. The author then decided on how each of the different animations could link together in a way that would feel seamless to the user, leading from one room to another smoothly. Once each of the animations was rendered, they were imported into the Flash library, where Actionscript 3 was used to program arrows to appear at the end of each animation, in order to allow the user to navigate to through the next are of the model. Each animation was given an object reference, which would then be linked to an arrow which would appear before the user, in order to allow them to make their choice, Fig 7.5 shows an example of this.

7.4 Components design with the Principles of HCI in mind

As mentioned in §5.1.4 the main aspect of the website, the 3D virtual tour, was designed to sit in the centre of the screen (Fig 7.1), with everything else surrounding it. This was to allow the users to quickly see the animation, and easily select which area they would like to view. Directly underneath the animation window, lies a 2D map which was synchronised with the users' position in the 3D world. This was designed to help users understand their current position and direction. Additional textual information with descriptions of what the user is looking at is positioned to the right of the animation, this would automatically update, as the user navigated through the 3D world. If the user required more time to read the information given,

Figure 7.5: Example to show navigational arrows

they can simply pause the animation, and resume playback once they are ready. The virtual tour was designed in such a way, that would allow the users to navigate the entire area, without getting lost or confused, which was achieved by giving the user, the appearance of having some sort of control over where they are travelling, while being restricted to the accurate direction of the virtual tour. Other design approaches incorporated a search function (which allows the user to scrub through the animation) combined with automatic navigation from current to target place, were placed in the form of quick navigation buttons, positioned to the left of the animation. Secondary interactive areas embedded into the website were given a less prominent position on the web page. However buttons that were easy to see and change colour upon mouse-over and selection were used to give the user a sense of visual feedback. Everything was designed to fit on a single page, hyperlinks and access points were kept to a minimum throughout the page [God02] and the entire site was designed to allow interaction using basic mouse input [KMB93] of pointing and clicking (§5.1.3).

A help page was also included as further instruction, that incorporated images of how the user could navigate around the website and which elements were interactive.

7.5 Testing program, table results and explanations

In order to speed up loading times, and to find out the optimum level of detail needed on a 3D object with various shaders and materials applied, tests were run on various machines and servers in order to ascertain which setup would be best used under certain conditions.

7.5.1 What was tested?

The Following experiments were designed to test the speed and efficiency of PV3D over the web. These included tests that calculated the frame rate (FPS), memory usage, as well as the impact of the user experience when LOD was greatly increased. The tests were split into two experiments, the first §7.5.1 was to test the impact of FPS when using different shaders on a model. In the interests of a fair experiment, the same tests were run from two different servers in different parts of the world. This was used to calculate the download times of each shader from the alternative servers. The second test §7.5.1 was designed to calculate the memory usage and frame rate of an increasingly detailed object. These were conducted on three separate machines of varying power, to establish if users with old computers could still receive a positive experience without any drop in quality (i.e high FPS and low memory usage). Models could be preloaded with more polygons and used to navigate locally; however these experiments where made to test how instantaneous PV3D could be over the web, without having the need to wait for each object to be downloaded over varying Internet connection speeds.

Experiment type 1

Initially a separate web page (Fig.7.6) was created that contained links for each individual shader type. This was hosted on two different servers, one based in Maidenhead, UK (Server 1) and the other in Minneapolis, USA, which trace-routes back to Tewkesbury, UK (Server 2). Another webpage that contained all shaders together on a single page was also hosted on Server 1 and Server 2. These were used to test the download time (DT) of each shader from the servers at different times of the day for 4 days; an average was taken and put into a table.

Figure 7.6: Screenshots of webpage for Experiment 1

After noticing certain consistencies in the results shown in Tables 7.1 - 7.4. The author decided to expand on the experiments by investigating how detailed a model could be without loss of visual quality in terms of smoothness (FPS) and memory usage.

Experiment type 2

An extension of this experiment was then done by creating a web page that contained a LOD object with Compound Material shader applied (Fig.7.7), that could be manipulated in real-time by clicking a mouse button to increase the polygon count of the model. The FPS and memory used were recorded on different machines varying in processing power, these ranged from an Intel Celeron D, 2.9Ghz (Linux OS); Intel Core 2 Duo, 2.1Ghz(Windows OS); and an Intel Core 2 Quad, 2.4Ghz (Windows OS).

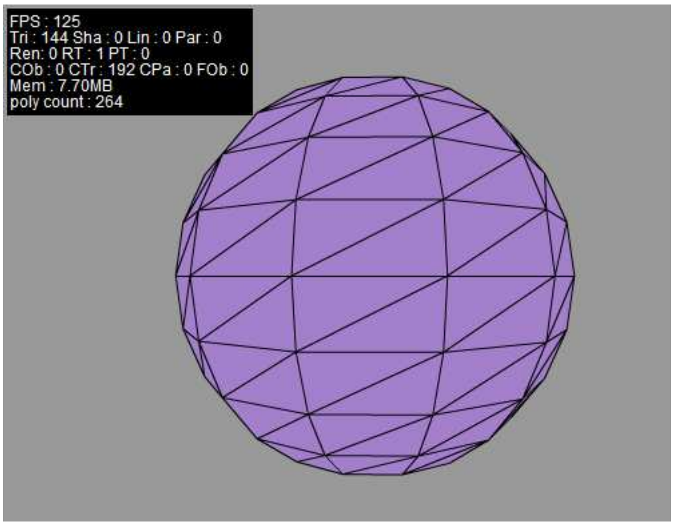

Figure 7.7: Screenshot of Experiment 2, LOD

7.6 How the results were recorded

Initially a Firefox browser extension plugin Firebug with addon Yslow was used to find out file sizes and download times, however this proved to be slightly unscientific and very inefficient; therefore it was decided that scripts were needed to automate the process. Experiment 1 (§7.5.1), needed to be conducted at the same time of day for 4 days on different servers. In order to do this, a shell script file (Listing 7.2) was created to locate and download each model with its own shader at the specified web addresses. Once this was completed, each address was processed through a Perl script

(Listing 7.3) which was used to calculate the download times and file sizes of each object, and rearrange and output all the data to a spreadsheet. In order for the shell script to be executed at exactly the right time of each day, a "cron job" (Listing 7.1) was set in the linux environment. This is a single line of code, telling the operating system to run a specified file at a certain time.

Listing 7.1: Cron Job used to run .sh file at certain times of the day

```
%.---------------- minute (0 - 59)
%|  .------------- hour (0 - 23)
%|  |  .---------- day of month (1 - 31)
%|  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
%|  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7)
%|  |  |  |  |
%*  *  *  *  *  command to be executed

0 10  *  *  *  /home/path_directory/shellscript.sh
0 18  *  *  *  /home/path_directory/shellscript.sh
```

Listing 7.2: Shell Script

```
#!/bin/bash

LINKS=( 'http://www.thierryzoghbi.co.uk/thesis/flatMatshader/flatTest.swf' 'http://www.thierryzoghbi.co.uk/thesis/phongMatshader/phongTest.swf' 'http://www.thierryzoghbi.co.uk/thesis/cellMatshader/CellTest.swf' 'http://www.thierryzoghbi.co.uk/thesis/colourMatshader/clrTest.swf' 'http://www.thierryzoghbi.co.uk/thesis/compMatshader/compTest.swf' 'http://www.thierryzoghbi.co.uk/thesis/envMatshader/MaterialTest.swf' 'http://www.thierryzoghbi.co.uk/scaleONLINE/scale.swf')

ELEMENTS=${#LINKS[@]}


  for ((i=0; i < $ELEMENTS; i++)); do
      ./shellscript2.pl ${LINKS[${i}]} --append-output=myserverdata.txt

      done


LINKS2=( 'http://www.urbanmodellinggroup.co.uk/TZ/flatMatshader/flatTest.swf' 'http://www.urbanmodellinggroup.co.uk/TZ/phongMatshader/phongTest.swf' 'http://www.urbanmodellinggroup.co.uk/TZ/cellMatshader/CellTest.swf' 'http://www.urbanmodellinggroup.co.uk/TZ/colourMatshader/clrTest.swf' 'http://www.urbanmodellinggroup.co.uk/TZ/compMatshader/compTest.swf' 'http://www.urbanmodellinggroup.co.uk/TZ/envMatshader/MaterialTest.swf' 'http://www.urbanmodellinggroup.co.uk/TZ/scaleONLINE/scale.swf')
```

```
ELEMENTS2=${#LINKS2[@]}


  for ((i=0; i < $ELEMENTS2; i++)); do
      ./shellscript2.pl ${LINKS2[${i}]} --append-output=umgserverdata.txt

  done
```

Listing 7.3: Perl Script

```
#!/usr/bin/env perl

use strict;
use warnings;

use LWP::UserAgent;
use Time::HiRes qw(time sleep tv_interval);

my ($browser, $request, $response) = undef;
my ($start_time, $stop_time, $time_delta) = undef;

if (not $ARGV[0]) {
    printf STDERR "ERROR: The target URL is missing; please specify.\n";
    exit(255);    # Non-zero exit code.
}

$browser = new LWP::UserAgent;
$browser->agent('Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.1)'); #options where chosen to solve bug issues
$request = new HTTP::Request('GET', $ARGV[0]);
$start_time = time();
$response = $browser->request($request);
$stop_time = time();

if ($response->is_success) {
    $time_delta = tv_interval([$start_time],[$stop_time]);
    printf STDOUT "%i %.2f %.3f\n", $stop_time, $time_delta, ($time_delta * 1000);
} else {
    printf STDERR "ERROR: %s\n", $response->status_line;
    exit(1);    # Something more sensible for an error.
}
```

7.6.1 Results tables and graphs

Experiment 1 was designed to find the download times (DT) and file sizes of each shader type on the same 3D object. To compensate for internet traffic, the same experiment was run twice a day, for four days. This was designed to run the test at peak and off-peak times in order to see if download times of the different file sizes would be affected. After finding that there was not such a big difference between the download times of any of the shaders on either server, the author decided to see whether the polygon count would effect the quality of the user experience (Experiment 2). This was performed with basic (Intel Celeron, 2.9Ghz CPU), medium (Intel Core 2 Duo, 2.1Ghz CPU) and high (Intel Core 2 Quad, 2.4Ghz CPU) specification computers.

Day 1				
10am				
Server 1 (my Server)				
	Shader Type	**File size (kb)**	**DT (sec)**	**DT (ms)**
	Flat Material Shader	113.4	0.77	774.932
	Phong Material Shader	113.9	0.73	728.66
	Cell Material Shader	113.9	0.85	854.876
	Colour Material Shader	112.1	0.81	810.78
	Compound Material Shader	112.2	0.71	709.921
	Environmental Map Material Shader	113.2	0.72	723.282
	All shaders together	337	1.65	1652.058
Server 2 (UMG server)				
	Flat Material Shader	113.4	1.4	1399.698
	Phong Material Shader	113.9	1.55	1550.682
	Cell Material Shader	113.9	1.35	1351.068
	Colour Material Shader	112.1	2	2000.633
	Compound Material Shader	112.2	1.26	1261.508
	Environmental Map Material Shader	113.2	1.62	1615.062
	All shaders together	337	2.92	2921.935
6pm				
Server 1 (my Server)				
	Flat Material Shader	113.4	7.4	7398.684
	Phong Material Shader	113.9	0.75	748.469
	Cell Material Shader	113.9	0.72	719.241
	Colour Material Shader	112.1	0.7	700.718
	Compound Material Shader	112.2	0.69	691.223
	Environmental Map Material Shader	113.2	0.7	704.63
	All shaders together	337	1.56	1556.996
Server 2 (UMG server)				
	Flat Material Shader	113.4	1.84	1844.271
	Phong Material Shader	113.9	4.48	4481.712
	Cell Material Shader	113.9	1.72	1718.466
	Colour Material Shader	112.1	2.59	2585.544
	Compound Material Shader	112.2	4.16	4157.885
	Environmental Map Material Shader	113.2	2.29	2292.461
	All shaders together	337	3.14	3140.392

Table 7.1: Experiment 1: Day 1 Results

Day 2				
10am				
Server 1 (my Server)				
	Shader Type	**File size (kb)**	**DT (sec)**	**DT (ms)**
	Flat Material Shader	113.4	0.93	934.056
	Phong Material Shader	113.9	0.62	620.339
	Cell Material Shader	113.9	0.63	625.707
	Colour Material Shader	112.1	0.63	627.334
	Compound Material Shader	112.2	0.69	687.221
	Environmental Map Material Shader	113.2	0.62	618.944
	All shaders together	337	1.6	1599.711
Server 2 (UMG server)				
	Flat Material Shader	113.4	0.87	866.469
	Phong Material Shader	113.9	1.13	1131.771
	Cell Material Shader	113.9	1.29	1289.496
	Colour Material Shader	112.1	1.35	1346.099
	Compound Material Shader	112.2	1.2	1197.892
	Environmental Map Material Shader	113.2	1.33	1328.446
	All shaders together	337	3.07	3074.108
6pm				
Server 1 (my Server)				
	Flat Material Shader	113.4	1.42	1419.799
	Phong Material Shader	113.9	0.92	915.42
	Cell Material Shader	113.9	0.99	986.473
	Colour Material Shader	112.1	0.79	794.466
	Compound Material Shader	112.2	0.87	873.479
	Environmental Map Material Shader	113.2	0.99	986.746
	All shaders together	337	1.76	1763.929
Server 2 (UMG server)				
	Flat Material Shader	113.4	1.6	1602.593
	Phong Material Shader	113.9	1.9	1898.169
	Cell Material Shader	113.9	2.16	2159.234
	Colour Material Shader	112.1	1.9	1903.621
	Compound Material Shader	112.2	2.25	2254.419
	Environmental Map Material Shader	113.2	1.32	1319.765
	All shaders together	337	4.09	4091.178

Table 7.2: Experiment 1: Day 2 Results

Day 3				
10am				
Server 1 (my Server)				
	Shader Type	**File size (kb)**	**DT (sec)**	**DT (ms)**
	Flat Material Shader	113.4	0.7	702.305
	Phong Material Shader	113.9	0.65	652.038
	Cell Material Shader	113.9	0.67	670.885
	Colour Material Shader	112.1	0.65	654.73
	Compound Material Shader	112.2	0.65	647.978
	Environmental Map Material Shader	113.2	0.66	662.731
	All shaders together	337	1.53	1530.995
Server 2 (UMG server)				
	Flat Material Shader	113.4	1.42	1420.655
	Phong Material Shader	113.9	1.16	1164.267
	Cell Material Shader	113.9	1.26	1255.748
	Colour Material Shader	112.1	1.2	1196.141
	Compound Material Shader	112.2	1.53	1533.313
	Environmental Map Material Shader	113.2	1.82	1819.804
	All shaders together	337	4.16	4160.651
6pm				
Server 1 (my Server)				
	Flat Material Shader	113.4	0.99	985.785
	Phong Material Shader	113.9	0.73	731.021
	Cell Material Shader	113.9	0.78	784.126
	Colour Material Shader	112.1	0.74	742.043
	Compound Material Shader	112.2	0.85	846.22
	Environmental Map Material Shader	113.2	0.7	702.204
	All shaders together	337	1.57	1565.772
Server 2 (UMG server)				
	Flat Material Shader	113.4	1.74	1736.118
	Phong Material Shader	113.9	1.6	1599.409
	Cell Material Shader	113.9	2.12	2116.708
	Colour Material Shader	112.1	1.92	1915.619
	Compound Material Shader	112.2	1.77	1771.182
	Environmental Map Material Shader	113.2	1.76	1762.849
	All shaders together	337	4.16	4157.62

Table 7.3: Experiment 1: Day 3 Results

Day 4				
10am				
Server 1 (my Server)				
	Shader Type	**File size (kb)**	**DT (sec)**	**DT (ms)**
	Flat Material Shader	113.4	0.67	672.933
	Phong Material Shader	113.9	0.76	759.171
	Cell Material Shader	113.9	0.68	682.202
	Colour Material Shader	112.1	0.78	784.51
	Compound Material Shader	112.2	0.77	766.024
	Environmental Map Material Shader	113.2	0.75	747.332
	All shaders together	337	1.58	1576.064
Server 2 (UMG server)				
	Flat Material Shader	113.4	1.63	1629.587
	Phong Material Shader	113.9	1.38	1377.683
	Cell Material Shader	113.9	1.44	1439.507
	Colour Material Shader	112.1	1.57	1567.341
	Compound Material Shader	112.2	1.57	1567.561
	Environmental Map Material Shader	113.2	6.7	6701.239
	All shaders together	337	4.37	4365.096
6pm				
Server 1 (my Server)				
	Flat Material Shader	113.4	0.74	742.15
	Phong Material Shader	113.9	0.72	716.904
	Cell Material Shader	113.9	0.68	682.762
	Colour Material Shader	112.1	0.74	737.975
	Compound Material Shader	112.2	0.69	691.027
	Environmental Map Material Shader	113.2	0.68	680.251
	All shaders together	337	1.57	1565.366
Server 2 (UMG server)				
	Flat Material Shader	113.4	1.35	1349.492
	Phong Material Shader	113.9	1.59	1586.118
	Cell Material Shader	113.9	1.63	1627.648
	Colour Material Shader	112.1	6.43	6432.908
	Compound Material Shader	112.2	1.91	1912.534
	Environmental Map Material Shader	113.2	1.67	1673.091
	All shaders together	337	4.81	4812.249

Table 7.4: Experiment 1: Day 4 Results

Celeron D Machine				
Polycount	**Render Time (ms)**	**Min FPS**	**Max FPS**	**Max Memory Used (MB)**
1012	2	14	25	6.02
2012	4	14	18	7.59
3120	6	12	14	23.2
4140	8	8	11	59.8
5100	11	8	10	91.6
6160	15	7	8	114.3
7080	18	6	7	145.2
8064	20	5	7	182.6
9112	22	5	6	218.3
10024	22	5	6	275.1
20200	40	2	2	736.8

Table 7.5: Experiment 2: Celeron D Machine

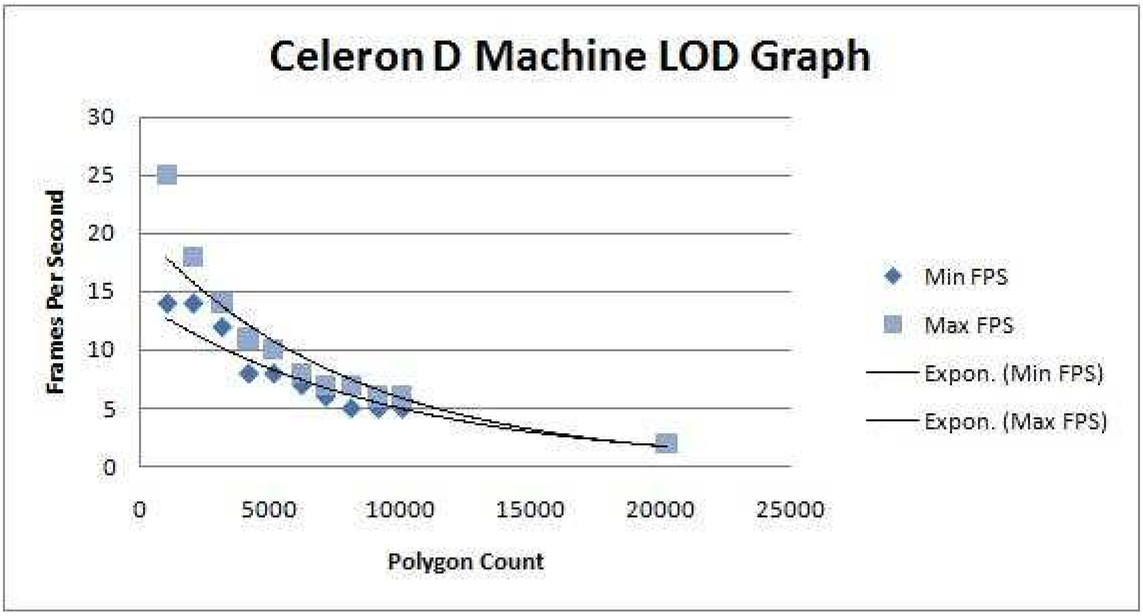

Figure 7.8: Celeron D Machine Results Graph

Core 2 Machine				
Polycount	**Render Time (ms)**	**Min FPS**	**Max FPS**	**Max Memory Used (MB)**
1012	2	28	52	6.1
2012	3	23	43	7.9
3120	4	17	35	9.5
4140	4.3	11	28	35.8
5100	6	23	30	49.3
6160	6.7	20	26	82.6
7080	7	20	24	116.8
8064	7.2	18	29	157.4
9112	13	15	20	181.3
10024	13	13	17	227.8
20200	18	6	11	729.6

Table 7.6: Experiment 2: Core 2 Duo Machine

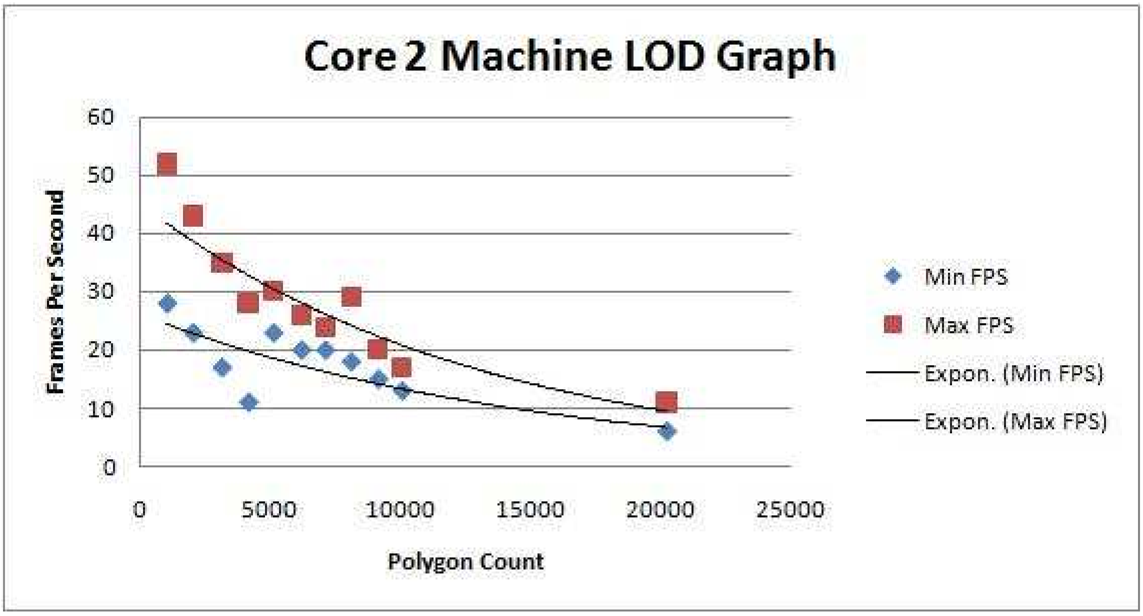

Figure 7.9: Core 2 Machine Results Graph

Quad Core Machine				
Polycount	**Render Time (ms)**	**Min FPS**	**Max FPS**	**Max Memory Used (MB)**
1012	0.5	58	61	6
2012	1.5	58	61	7.9
3120	2.5	46	55	9.9
4140	3	38	44	11.8
5100	4	31	38	32.4
6160	5	29	33	70.9
7080	5.5	23	29	104.4
8064	7	24	27	138.2
9112	9	19	24	166.2
10024	13	19	23	209.9
20200	20	9	13	714.6

Table 7.7: Experiment 2: Quad Core Machine

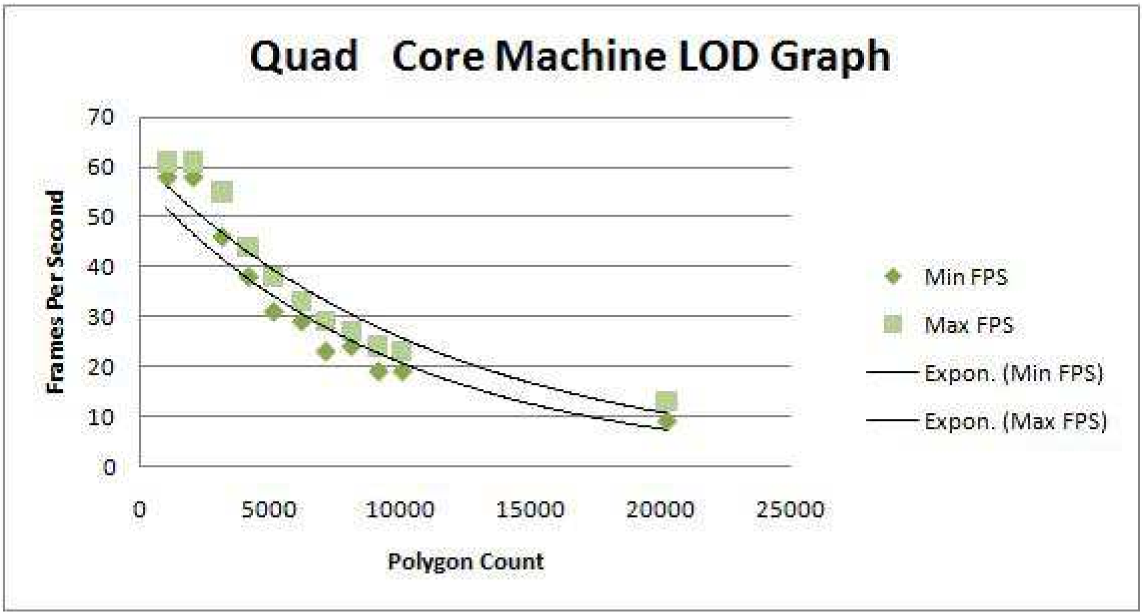

Figure 7.10: Quad Core Machine Results Graph

7.7 Conclusion

As seen from the results, download times of objects including their shaders is fairly minimal, and users with a standard internet connection (1Mb/s) at present will not need to wait, as loading appears almost instantly. Although PV3D is graphics card independent, the processor is still taxed. While running a lower polygon count (less than 5000), all processors seemed to cope without any loss of FPS, however when the polygon count was increased (greater than 5000) the Celeron D machine showed decreasing visual quality (Table 7.8). However when pushed to the limit of 20,000 polygons, all machines could not keep a suitable frame rate of 15FPS or more. This demonstrated the fact that PV3D does not satisfactorily support objects greater than 12,000 polygons, even with a powerful machine, showing that it still has its limitations.

Chapter 8

Investigation

8.1 Introduction

After creating the interface for displaying 3D content, It was necessary to find out whether the interface designed, allowed people of all ages and computer abilities to experience historical buildings and their "artifacts" within their own home. This chapter will show and discuss the results of the questionnaire using charts and graphs. The questionnaire was emailed and handed out to a wide sample audience (roughly around 30 people of varying ages and computer abilities) Bias was avoided by including people that the author did not directly know. To maintain a high quality questionnaire, questions were placed in a logical sequence to give the participant a better mental map as he/she is filling out the questionnaire [BSW04] as well as considering question conciseness and length [CBB+98]. A sample questionnaire was distributed to 5 people [SRP02] in order to attain feedback in relation to the quality of the questionnaire; whether they had understood the main points [BSW04] and if they had any difficulties with the questions [PCL+04]. Once this was done, the original questionnaire was slightly amended and the final copy as shown in §8.2 was produced.

8.2 Questionnaire

St Andrews Hall Application Questionnaire

Please answer this questionnaire after you have visited the website application as directed in the email. If you are printing this to return it by hand, please use black or blue ink and write clearly in BLOCK CAPITALS.

Leave blank if you wish to remain anonymous.

Name:...

Please tick the correct box.

Age: ☐ 10 – 19 years ☐ 20 – 29 years ☐ 30 – 39 years ☐ 40 – 49 years ☐ 50 + years

Computer (IT) Literacy: ☐ Very IT literate ☐ Somewhat IT literate ☐ Not IT literate

Were you using a broadband connection when using the application? ☐ Yes ☐ No

How easy was it to navigate around the application?
☐ Very Simple ☐ Simple ☐ Difficult ☐ Very Difficult

Did you find the Help Screen helpful?
☐ Yes ☐ No ☐ Didn't need to use it

In your opinion, how quickly did the application load?
☐ Very Quickly ☐ Quickly ☐ Slowly ☐ Very Slowly

Would you have enjoyed audio output to aid the animations?
☐ Yes ☐ No ☐ Maybe

Did you manage to learn anything about St Andrews Hall in the process?
☐ Yes ☐ No

Did you find that using the mouse and keyboard made interacting with the application easier?
☐ Yes ☐ No

Are there any other comments you would like to add about your experiences using the St Andrews Hall Application?

....

....

....

....

....

8.3 Findings and Results

The investigation received a total of 17 completed replies from the participants, a break down analysis of the results are shown below. It is important to note that 100% of all participants had access to broadband and found the application very simple to navigate using their mouse and keyboard.

8.3.1 Age analysis:

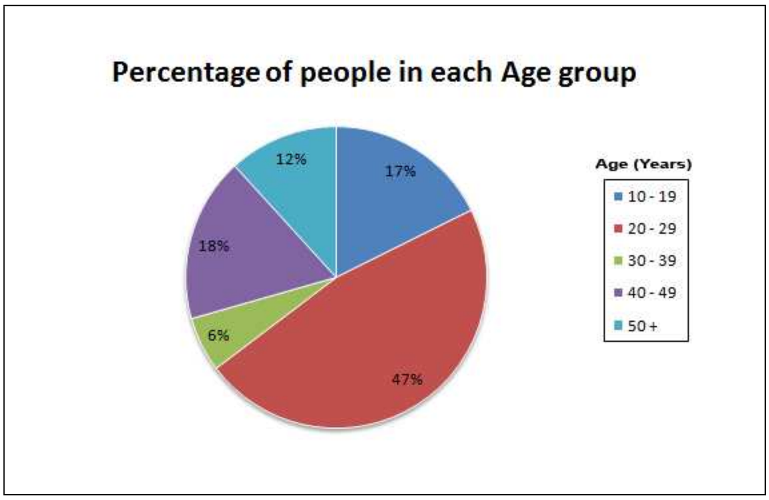

Figure 8.1: A graph to show the Percentage of people in each age group who completed the questionnaire.

Even though the greatest portion of participants where in their twenties, figure 8.1 shows a diversity of age in the respondents to the questionnaire.

8.3.2 IT literacy analysis:

It was necessary to find out how users rated their levels of IT competency/literacy in order to see whether the application could easily be used by people who are not confident using a computer. Figure 8.2 shows that four of the participants classified themselves as being IT illiterate, with the majority of users only being somewhat IT literate.

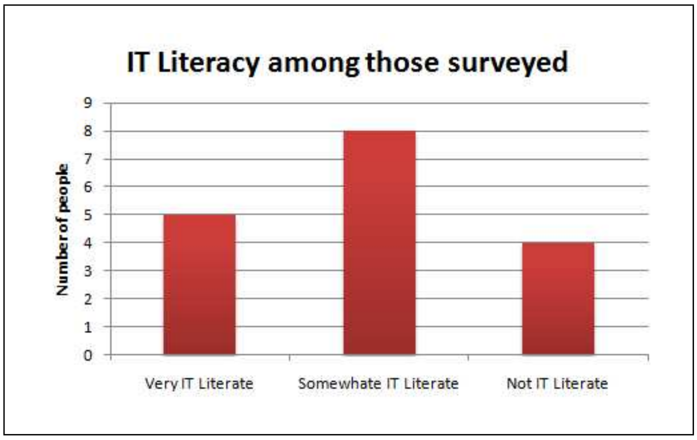

Figure 8.2: A graph showing the IT Literacy of the participants.

8.3.3 Application Simplicity analysis:

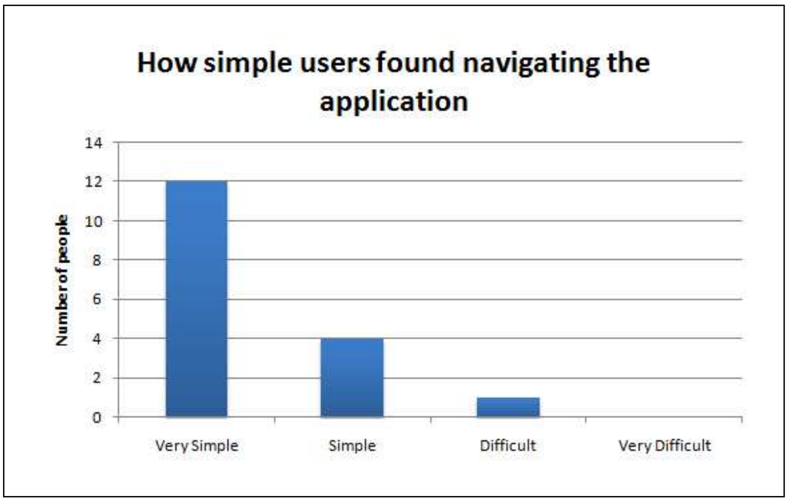

Figure 8.3: A graph showing the applications navigational simplicity.

Having seen from figure 8.3, only one participant out of the possible seventeen found it difficult to navigate the application. This number was expected to be higher as in figure 8.2, four users had rated themselves as being non IT literate. This shows that the application had reached its aim of making a simple user interface, while allowing the use of a standard mouse and keyboard (Fig.8.4) that participants are already familiar with.

As noted in §8.3, 100% of all applicants were using a broadband connection,

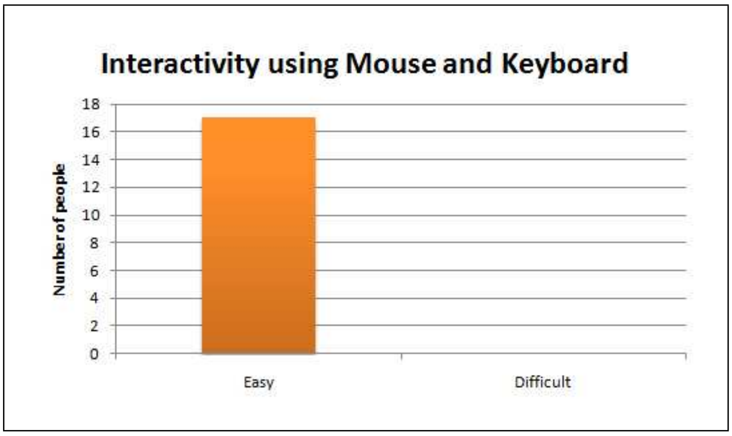

Figure 8.4: A graph showing the participants capability of navigating the application using a standard keyboard and mouse.

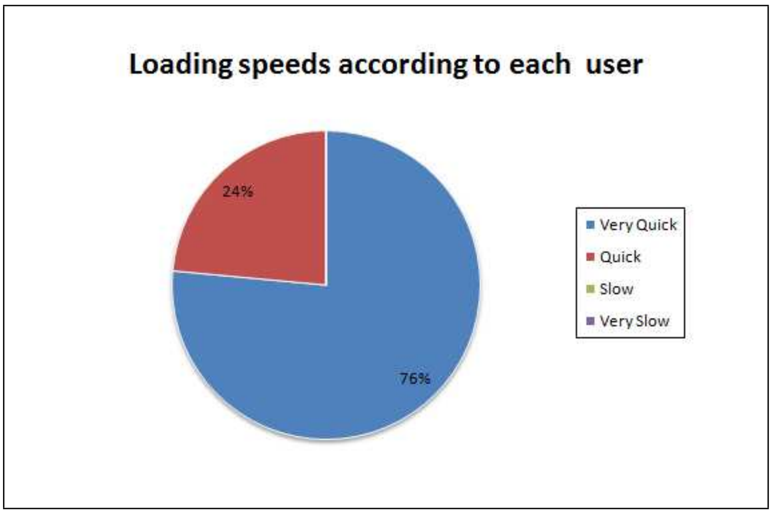

Figure 8.5: A graph showing the loading speeds of the application.

therefore loading speeds was not a problem as four participants found the loading times in their opinion, to be quick and thirteen found the loading times to be very quick.

8.3.4 Expandability analysis:

From the initial sample questionnaire, some users voiced an idea of audio as an additional output to the animations, as they felt that it was slightly difficult to read and watch the video being played at the same time. This resulted in an additional question being asked in the final questionnaire. Figure 8.6 shows that the majority of users would have liked to have some form of audio to accompany the animations.

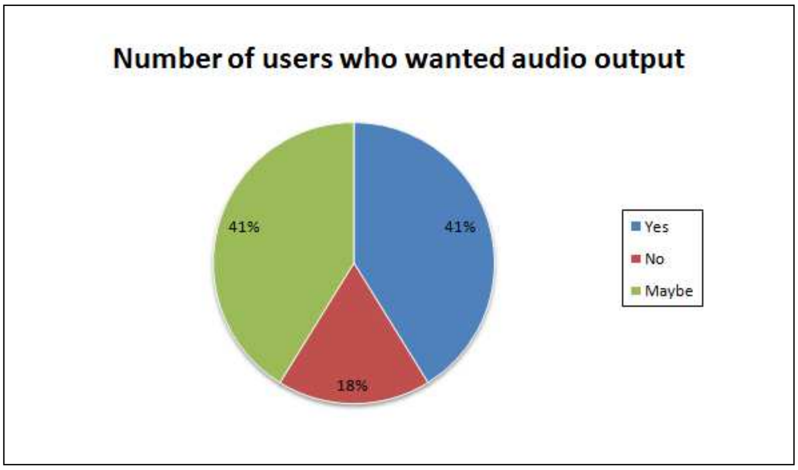

Figure 8.6: A graph showing the percentage of users who would have liked audio

The additional comments proved to be popular with users, giving them a chance to explain what they liked and didn't like so much about the application. Most people found the Augmented Reality section to be very entertaining and had asked if it where possible to expand and allow it to show other objects. A few were not so keen about the navigational map at the bottom of the application suggesting it was a bit misleading as the red dot did not follow the path of the video, but remained static.

8.4 Evaluation

From the investigations conducted, it was clear that the aim of creating a simple user interface to access 3D Cultural Heritage buildings was achieved. However there are future improvements that should be addressed; including necessary audio output for the application and the expandability into the real-time environments that users are becoming accustomed to. Chapter 9 discusses this in further detail.

Chapter 9

Conclusion and future work

Over the course of this thesis a wide range of techniques were investigated and implemented to enable the exploration and interaction with St Andrews Hall and its contents. The entire web based application was designed for control with mouse clicks and gestures as this would reach a much broader audience over the internet, as every user is accustomed with a mouse or touch pad as the basic form of input to their computer.

As all animations for the virtual tour were pre-rendered, it was not necessary to think of any form of LOD or culling techniques, which would otherwise have been critical in a real time application. Therefore the buildings were modelled using historical references and photographs of similar structures from the same period. Attention to detail was paramount in order to get an accurate representation of the cloisters in the early 18th century. While little needs to be mentioned about the process of modelling the cloisters, as Chapter 4.3 was dedicated to it, it was a successful indication of how high quality 3D virtual historical reconstruction can be used for educational and archive purposes. The UEA History department are pleased with the overall results, as everything was kept to their (correct) historical requirements.

9.1 Flash and Interactivity

The aim of this thesis was to design an interactive user interface that would reach a broad range of people, including those that regard themselves as computer illiterate; therefore the main focus had to be ease of use. Flash was investigated and seeing as it has already been widely accepted as a web browser standard, and with most common browsers already incorporating Flash into their API, there was no need for any user to download and install extra software to view and interact with the website application.

As Flash can be stored and run on a local machine, as well as the internet, it would also be an easy transition from online website to an in-house museum kiosk. Flash was also chosen due to its nature which allowed many plugins and add-ons that could be used to import 3D models from 3D modelling packages, such as 3D Studio Max. While Flash is a WYSIWYG (what you see is what you get) editor, it also has a powerful script in the form of Actionscript which allows for detailed manipulation of objects, images and video files. This flexibility and industry wide use made it a clear choice in designing an interactive virtual tour web site.

The choice to design a semi real time application was based on the fact that a lot more detail could theoretically be introduced into the modelling, as well as giving the user, one less thing to think about while navigating the area. It would also eliminate any form of confusion while travelling along the virtual building, with all areas being labelled, and each section shown in its entirety. Not only does this technique portray control to the user, it also allows for a more controlled interaction without loss of immersiveness.

Overall the methods used to implement the aim of this thesis worked well as various interactive ideas were implemented within a single webpage, that was easy to use and navigate.

9.2 Papervision3D

When choosing how to import 3D models into the Flash environment, there were plenty of packages that were appealing to use. However, the main contenders were narrowed down to the open source packages mainly because they were free to download and implement. Papervision3D uses the popular COLLADA (see §7.2) extension format to export models from the 3D Studio Max package and import them into the Flash environment.

Papervision was extensively tested for use with different servers, computers, polygon sizes and various textures and shaders. Therefore both hardware and software alternatives were varied in order to establish an optimal solution. It is important firstly to note the specifications of each of the computers on which the various tests were undertaken, so that a more accurate comparison of the results can be seen. The specifications are as follows:

Celeron D Machine:

CPU: Intel Celeron D 2.9Ghz

Graphics: Integrated Intel GMA X3000 Dynamic Video Memory

Ram: 1GB (DDR2)

OS: Linux Ubuntu 9.0

Core 2 Duo Machine:

CPU: Intel Core 2 Duo 2.1Ghz

Graphics: Nvidea GeForce 8600M GT

Ram: 3GB (DDR2)

OS: Windows Vista 32bit

Quad Core Machine:

CPU: Intel Core 2 Quad 2.4Ghz

Graphics: Nvidea GeForce 8800 GT

OS: Windows Vista 64bit

At the time, the majority of the work on these implementations took place mid 2008, where the setups used were about average in terms of home computers, from a few years old to current specification. Although the Core 2 Duo machine is a mid range set up in this experiment, it would loosely be classified as a mid-high range machine as it is a laptop, and not a desktop. This allowed for fair testing of the application across a broad range of processor power.

While conducting the Papervision experiments, it was seen that Papervision is graphics card independent, as it relies more heavily on the CPU for its calculations. Therefore users with on board graphics cards (not especially powerful graphics) can easily run basic real time 3D objects (less than 10,000 polygons) and interact with them as long as they have an average off the shelf CPU (2Ghz clock speed) as their processor.

However, it was found that no matter how powerful the CPU (tested with 4 core processors at 2.4Ghz), Papervision will produce less than amicable results when it comes down to a large polygon amount. Using an average machine with 2 cores at around 2Ghz or more, allows the ability to process up to 10,000 polygons with a satisfactory frame rate of 20 frames per second. However, with single core processors (ranging between 2Ghz and 3Ghz, this amount is limited to a maximum of roughly 4000 polygons. Both results also include an added shader.

The file sizes for objects and shaders of the COLLADA file type are kept fairly minimal, with an average file size of 150kb for an object between 2000 and 5000 polygons. Therefore download and loading times are kept very minimal, with no adverse effect on the user, such as overall bandwidth usage or time waited for complete load.

The experiments also showed that shaders did not have a negative effect on loading times or running quality of the 3D objects, however, the more objects and shaders that were added to the scene, the more powerful a processor was needed to keep everything running smoothly.

9.3 Research Expandability

This thesis looked into many forms of online interactive content for 3D objects and environments, however most of the interactivity was conducted through the use of a mouse. As technology progresses, and user home computers become more and more powerful, it would be a good idea to look into alternate ways of interaction, using alternative methods of input, such as speech and gesture recognition using a digital web-cam, as well as 6DOF devices that allow the user to "feel" around specific objects. Virtual Reality head sets can also be added so that the users can feel fully immersed into their surroundings with 180 degrees of undisturbed viewing.

Augmented reality can be investigated not only to display 3D objects, but to play animations and videos, this would be an example that would allow users to pick up a certain piece of card, and be transported to the location that is associated with the picture on display. As rendering capabilities and home computer systems improve, there is no limit to the detailed realism of 3D models and environments, that will one day be required to be rendered in real time, with complete user interaction.

Historical reconstruction as well as virtual modelling can have a huge impact on useful and stimulating applications now and far into the future, with the continuing positive advancement in technology, the possibilities are endless.

Bibliography

[20110] VAST 2010. 11th vast international symposium on virtual reality, archeology and cultural heritage, [accessed October 2010]. http://vast2010.org/.

[3Dc] 3Dconnexion. Spacenavigator. http://www.3dconnexion.com/products/spacenavigator.html.

[AADH+08] D. Arnold, J. Glauert A. Day, S. Haegler, V. Jennings, B. Kevelham, R. Laycock, N. Magnenat-Thalmann, Jonathan Maim, D. Maupu, G. Papagiannakis, D. Thalmann, Barbara Yersin, and K. Rodriguez-Echavarria. Tools for populating cultural heritage environments with interactive virtual humans. *EPOCH Conference on Open Digital Cultural Heritage Systems*, 2008.

[Aca10] Dunhuang Academy. Mogao cave 322, [accessed August 2010]. http://enweb.dha.ac.cn/001F/index.htm.

[AJS97] Accot, Johnny, and Zhai Shumin. Beyond fitts' law: models for trajectory-based hci tasks. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 295–302. ACM, 1997.

[amgaM10] Negroponte's architecture machine group at MIT. Put that there, 1981, (accessed 2010). http://www.youtube.com/watch?v=PeWwfhsSqsc.

[Art10] John Enrique Arturo. An introduction to spline modeling, [accessed December 2010]. http://ezinearticles.com/?An-Introduction-to-Spline-Modeling&id=1560349.

[bas09] La basilica superiore, (accessed 2009). http://www.secundavita.com/gallery/9/images/basilica_superiore_JPG.jpg.

[BFW+01] Flack Willmott Browne, P. A. Flack, J. Willmott, S. P. Browne, D. B. Arnold, and A. M. Day. Scene assembly for large scale urban reconstructions. In *Proceedings VAST 2001*, pages 227–234, 2001.

[BKL+00] D Bowman, E Kruijff, J LaViola, M Mine, and I Poupyrev. An introduction to 3d user interface design. *ACM SIGGRAPH 2000, Course notes*, (36):200–0, 2000.

[Bol80] Richard A. Bolt. "put-that-there": Voice and gesture at the graphics interface. In *SIGGRAPH '80: Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pages 262–270, New York, NY, USA, 1980. ACM.

[Bor08] M. Borghi. Final project review report, [accessed September 2008]. http://mosaic.infobyte.it/project/report/final.html#_Toc442245222.

[BSW04] Norman M. BookBradburn, Seymour Sudman, and Brian Wansink. *Asking Questions: The Definitive Guide to Questionnaire Design For Market Research, Political Polls, and Social and Health Questionnaires.* Jossey-Bass, 2004.

[BTEHM98] P. Boulanger, J. Taylor, S. El-Hakim, and Marc. How to virtualize reality: An application to the re-creation of world heritage sites. *Proceedings of Fourth International Conference on Virtual Systems and Multimedia*, pages 39–45, 1998.

[BvDE+99] Judith R. Brown, Andy van Dam, Rae Earnshaw, Jose Encarnao, Richard Guedj, Jennifer Preece, Ben Shneiderman, and John Vince. Human-centered computing, online communities, and virtual environments. *IEEE Computer Graphics and Applications*, 19(6):70–74, 1999.

[BWW+01] S.P. Browne, J. Willmott, L.I. Wright, A.M. Day, and D.B. Arnold. Latest approaches to modelling and rendering large urban environments. In *EurographicsUK 2001*, 2001.

[CBB+98] Mick P. Couper, Reginald P. Baker, Jelke Bethlehem, Cynthia Z. F. Clark, Jean Martin, II William L. Nicholls, and James M. O'Reilly. *Computer Assisted Survey Information Collection*. Wiley, 1998.

[cit09] Procedural city engine, [accessed 2009]. `http://www.procedural.com/cityengine/features.html`.

[Cof10] 3D Coform. Tools and expertise for 3d collection formation, [accessed August 2010]. `http://www.3d-coform.eu/index.php/imprint`.

[col08] Collada, [accessed 2008]. `http://www.collada.org/mediawiki/index.php/Main_Page`.

[Con08] Web3D Consortium, [Online; accessed 30-October-2008]. `http://www.web3d.org/about/overview/`.

[Con10] Automatic Reconstruction Conduit. Automatic image based 3d reconstruction, [accessed September 2010]. `http://www.arc3d.be/`.

[cra] Crazybump. `http://www.crazybump.com/`.

[cry08] Virtual amsterdam crystal palace, [accessed 2008]. `http://www.paleisvoorvolksvlijt.nl/3d/`.

[DAHF04] A. M. Day, D. B. Arnold, S. Havemann, and D. W. Fellner. Combining polygonal and subdivision surface approaches to modelling and rendering of urban environments. *Computers Graphics*, 28(4):497 – 507, 2004.

[Dis09] Virtual Theatre District, [Online; accessed 25-January-2009]. http://artscool.cfa.cmu.edu/~hemef/pompeii.

[EGA+05] K. Echavarria, M. Genereux, D.B. Arnold, A.M. Day, and J.R.W. Glauert. Multilingual virtual city guides. In *International Conference Graphicon 2005*, 2005.

[Enc99] J. Encarnacao. New media for cultural heritage, 1999. Members of INI-GraphicsNet - Dramstadt University of Technology; Fraunhofer Institute for Computer Graphics; Fraunhofer Centre of Research in Computer Graphics USA; Centro de Computacao Grafica; Centre for Andvanced Media Technology.

[EPO09] EPOCH. European network of excellence in open cultural heritage, [accessed September 2009]. http://www.epoch-net.org.

[EV01] Rae Earnshaw and John Vince, editors. *Digital content creation*, page 354. Springer-Verlag New York, Inc., New York, NY, USA, 2001.

[fATitH08] The Institute for Advanced Technology in the Humanities. Tour ancient rome at siggraph 2008 in los angeles, 2008. Press Release.

[Fit54] P. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391, 1954.

[fla08] Virtual museum of the ancient via flaminia, [accessed 2008]. http://www.vhlab.itabc.cnr.it/flaminia/ricostruzioni_esplorazione.html.

[fla09] Flartoolkit, [accessed 2009]. http://www.libspark.org/wiki/saqoosha/FLARToolKit/en.

[FMS02] M. Farid, F. Murtagh, and J.L. Starck. Computer display control and interaction using eye-gaze. *Journal of the Society for Information Display*, 10:289–293, 2002.

[fre09] Freespin3d, [accessed 2009]. http://www.freespin3d.com/freespin3d/overview.html.

[Fri99] Kicki J. Frisch. The fully immersive cave. In *in H.-J. Bullinger O. Riedel, eds, '3. International Immersive Projection Technology Workshop, 10./11. May 1999, Center of the Fraunhofer Society Stuttgart IZS*, pages 59–63. Springer, 1999.

[FWB+01] P.A. Flack, J. Willmott, S.P. Browne, A.M. Day, and D.B. Arnold. Scene assembly for large scale reconstructions. In *Proceedings of VAST 2001*, Athens, 2001.

[GHvD94] M. Gleicher, K. Herndon, and A. van Dam. The challenges of 3d interaction. *SIGGHI Bulletin*, pages 36–43, 1994.

[God02] S Godin. *The Big Red Fez, How to make any website better.* Simon Schuster; 1st Fireside Ed edition, 2002.

[Gra03] Scintillating Graphics. Vrml magic, 2003. http://www.scintillatinggraphics.com.au/VRMLmagic/VRMLInfo.html.

[HH09] St Andrew's Hall and Blackfriars Hall. The halls, [accessed September 2009]. http://www.norwich1.com/st_andrew.htm#TheHalls.

[Hic52] W. E. Hick. On the rate of gain of information. *Quarterly Journal of Experimental Psychology*, 4:11–26, 1952.

[Hym53] R Hyman. Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology*, 45:188–196, 1953.

[inf] http://mosaic.infobyte.it.

[iso97] The virtual reality modeling language., 1997. http://www.web3d.org/x3d/specifications/vrml/ISO-IEC-14772-VRML97/.

[J.J99] J.Jacobson. The temple of isis 2d, 1999. http://planetjeff.net/isis/Isis-2D.html.

[Joh08] C. Johnson. Dudley castle, 1993 [Online; accessed 23-August-2008]. http://www.exrenda.net/dudley/index.htm.

[JV05] J. Jacobson and J. Vadnal. The virtual pompeii project. *World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education (E-learn), (Vancouver, Canada)*, 2005.

[KFW+79] J.H. Ten Kate, E.E.E. Frietman, W. Willems, B.M. Ter Haar Romeny, and E. Tenkink. Eye-switch controlled communication aids. In *Proceedings of the 12th International Conference on Medical Biological Engineering*, pages 19–20, 1979.

[KMB93] P Kabbash, I. S. MaxKenzie, and W Buxton. Human performance using computer input devices in the preferred and non-preferred hands. *ACM Human Factors in Computing Systems*, pages 474–481, 1993.

[KRB06] Verdy Kwee, Anthony Radford, and Dean Burton. Educative visuals - digital delivery of architectural informations for (potential) heritage buildings. *The 7th International Symposium on Virtual Reality, Archeology and Cultural Heritage*, pages 91–99, 2006.

[LD08] Drinkwater D. Laycock, R. G. and A. M. Day. Exploring cultural heritage sites through space and time. In *ACM Journal on Computing and Cultural Heritage*, pages Article 11, 11.9–11.10, October 2008.

[LLD06] R.G. Laycock, S.D. Laycock, and A.M. Day. Haptic navigation and exploration of high quality pre-rendered environments. *The 7th International Symposium on Virtual Reality, Archeology and Cultural Heritage*, pages 17–24, 2006.

[LSB+04] F. Liarokapis, S. Sylaiou, A. Basu, N. Mourkoussis, M. White, and P.F. Lister. Interactive visualisation interface for virtual museums. *The 5th International Symposium on Virtual Reality, Archeology and Cultural Heritage*, pages 46–56, 2004.

[Mac92] I.s. Mackenzie. *Fitts Law as a research and design tool in human-computer interaction.*, chapter 7, pages 91–139. 1992.

[Mar] Marker generator online. `http://flash.tarotaro.org/blog/2009/07/12/mgo2/#more-23`.

[McC04] Hugh McCabe. Interactive 2d and 3d graphics over the world-wide web with actionscript. *ITB Journal*, 10:4–24, 2004.

[MCO97] Yair Mann and Daniel Cohen-Or. Selective pixel transmission for navigating in remote virtual environments. *EUROGRAPHICS*, 16(3), 1997.

[MFMR09] V.T.C. Middleton, A. Fyall, M. Morgan, and A. Ranchhod. *Marketing in Travel and Tourism*, page 260. Butterworth-Heinemann, 2009.

[MWH+06] Pascal Mueller, Peter Wonka, Simon Haegler, Andreas Ulmer, and Luc Van Gool. Procedural modeling of buildings. *Proceedings of ACM SIGGRAPH 2006 / ACM Transactions on Graphics*, 25(3):614–623, 2006.

[MZS99] I. S. MacKenzie, S. X. Zhang, and R. W. Soukoreff. Test entry using soft keyboards. behaviour information technology. *Behaviour Information Technology.*, pages 235–244, 1999.

[NA09] Catholic Encyclopedia online New Advent. St. dominic, [accessed September 2009]. `http://www.newadvent.org/cathen/05106a.htm`.

[nor09a] Norwich blackfriars, [accessed 2009]. `http://www.norwichblackfriars.co.uk/watch/`.

[Nor09b] The Halls Norwich. History, [accessed September 2009]. `http://www.standrewshall.co.uk/History`.

[O3D09] Google o3d, [accessed August 2009]. `http://code.google.com/apis/o3d/docs/techoverview.html`.

[old08] Virtual old prague: Vop project, [accessed 2008]. http://www.cgg.cvut.cz/vsp/.

[onl09] The Norwich Blackfriars online. The halls and the reformation, [accessed September 2009]. http://www.norwichblackfriars.co.uk/the-halls-and-the-reformation/.

[PAC+01] D. Pape, J. Anstey, B. Carter, J. Leigh, M. Roussou, and T. Portlock. Virtual heritage at igrid 2000. *Proceedings INET 2001*, 2001.

[Pas09] Virtual Past. A university of east anglia enterprise, [accessed October 2009]. http://virtualpast.co.uk/.

[pau09] The papal basilica: St. paul outside-the-walls, [accessed 2009]. http://www.vatican.va/various/basiliche/san_paolo/vr_tour/index-en.html.

[PCL+04] Stanley Presser, Mick P. Couper, Judith T. Lessler, Elizabeth Martin, Jean Martin, Jennifer M. Rothgeb, and Eleanor Singer. *Methods for Testing and Evaluating Survey Questionnaires*. Wiley, 2004.

[pet09] Vatican necropolis, [accessed 2009]. http://www.vatican.va/various/basiliche/necropoli/scavi_english.html.

[PLJ+01] K. Park, J. Leigh, A. Johnson, B. Carter, J Brody, and J. Sosnoski. Distance learning classroom using virtual harlem. In *In the proceedings of the Seventh International Conference on Virtual Systems and Multimedia (VSMM 2001), Berkeley CA*, pages 489–498, 2001.

[PLW08] Thies Pfeiffer, Marc E. Latoschik, and Ipke Wachsmuth. Evaluation of binocular eye trackers and algorithms for 3d gaze interaction in virtual reality environments. *Journal of Virtual Reality and Broadcasting*, 5(16), December 2008. urn:nbn:de:0009-6-16605,, ISSN 1860-2037.

[PM01] Yoav I. H. Parish and Pascal Mueller. Procedural modeling of cities. In Eugene Fiume, editor, *Proceedings of ACM SIGGRAPH 2001*, pages 301–308, New York, NY, USA, 2001. ACM Press.

[Pro09a] Rome reborn, [accessed November 2009]. `http://www.procedural.com/showcase/showcases/rome-reborn.html`.

[Pro09b] Procedural. Cityengine product brochure. Online PDF, 2009. `http://www.procedural.com/fileadmin/user_upload/Medias/pdf/0905_CityEngine2009_Product_Brochure.01.pdf`.

[Res10] Microsoft Research. Project natal, xbox360, 2010. `http://www.xbox.com/en-US/live/projectnatal/`.

[rom08a] Aerial view of the entire city model, [accessed November 2008]. `http://www.romereborn.virginia.edu/rome_reborn_2_images/gallery/hi_res/RR1.0/Rome02.jpg`.

[Rom08b] Rome reborn, [accessed November 2008]. `http://www.romereborn.virginia.edu/index.php`.

[rom08c] Rome reborn press release, [accessed November 2008]. `http://www.romereborn.virginia.edu/rome_reborn_2_documents/project_news/google_earth_ancient_rome_release_final.pdf`.

[Rub92] B. Rubinstein. The micro gallery at the national gallery of london. *Archives and Museum Informatics*, 6(2):5–10, 1992.

[SC78] B. Burr S. Card, W. English. Evaluation of mouse, rate-controlled isometric joystick, step keys and text keys for text selection on a crt. *Ergonomics*, 21:601–613, 1978.

[Seo05a] S. Seow. Information theoretic models of hci: A comparison of the hick-hyman law and fitts' law. *Human-Computer Interaction*, 20:315–352, 2005.

[Seo05b] Steven C. Seow. Information theoretic models of hci: a comparison of the hick-hyman law and fitts' law. *Human-Computer Interaction*, 20:315–352, September 2005.

[SG72] G. Stiny and J. Gips. Shape grammars and the generative specification of painting and sculpture. *Information Processing*, 71:1460–1465, 1972.

[SRP02] H. Sharp, Y. Rogers, and J Preece. *Interaction Design: Beyond Human-Computer Interaction*. John Wiley & Sons, Inc., 2002.

[Sut77] Helen Sutermeister. *The Norwich Blackfriars*. East Anglian Printers, 1977.

[SW49] C.E. Shannon and W. Weaver. *The mathematical theory of communication*. Univ. Of Illinois Press, 1949.

[Tec10] Information Society Technologies. Epoch - excellence in processing open cultural heritage, [accessed October 2010]. `http://cordis.europa.eu/ist/digicult/epoch.htm`.

[UMG08] Urban modelling group, 2008. `http://urbanmodellinggroup.co.uk`.

[Vel99] K. Veltman. *Frontiers in Conceptual Navigation for Cultural Heritage*. Toronto: Ontario Library association, 1999.

[Vil09] Vilau. Tourism and cultural heritage, (Accessed online 2009). `http://www.vilau.net/data/docs/pdf_museosen_68427.pdf`.

[Wil08] J. Williams. 3d modelling and real time rendering techniques for complex urban environements. Master's thesis, The University of East Anglia, 2008.

[WWDA01] L.I. Wright, J. Willmott, A.M. Day, and D.B. Arnold. Rendering of large and complex urban environments for real time heritage reconstruction. In *Proceedings of VAST 2001*, Athens, 2001.

[Zar02] J. Zara. Concise tour to the virtual old prague. *Eurographics 2002*, 2002. http://www.cgg.cvut.cz/members/zara/papers/Zara-EG2002.pdf.

[Zar04] J Zara. Virtual reality and cultural heritage on the web. In *Proceedings of the 7th International Conference on Computer Graphics and Artificial Intelligence*, 2004.