

# Improving the experience of exploring a virtual museum

-  
Lund after 1658



Silvia Ramos Rodriguez



Master Thesis

DEPARTMENT OF DESIGN SCIENCES  
LUND UNIVERSITY, SWEDEN

Improving the experience of exploring  
a virtual museum

-

Lund after 1658

*Master thesis by*  
Silvia Ramos Rodriguez

*Supervisor*  
Mattias Wallergård

Dept. of Computer Science at Universitat Politècnica de Catalunya, Spain  
Performed at Dept. of Design Sciences at Lunds Universitet, Sweden

May 2009

ISRN: LUTMDN/TMAT-5126-SE  
EAT 2009

# Abstract

*Traditional museums are sometimes static, passive and lifeless. Based on previous work on interaction techniques in Virtual Reality museums, the aim of this Master Thesis is to bring this work to the next level. The project will focus on improving the experience of exploring a virtual museum through more realistic modeling and improved navigation and interaction with the objects allocated inside. As a novelty in this kind of virtual museum the interaction is based on the Nintendo Wii controllers -Wiimote and Nunchuck.*

*The Next Engine Desktop 3D Scanner is used to get base meshes for the objects, and Autodesk 3DStudio MAX and Blender are used as modeling tools. The high quality textures for the pictures exhibited in the virtual museum are created with Adobe Photoshop. We implement the navigation and interaction framework with EON Studio. Finally, the Bluetooth stack BlueSoleil and the emulator GlovePIE are the tools selected to connect the Wii controllers to the application. The visual appearance is improved with shadows, and the time of rendering is accelerated using render-to-texture techniques.*

*The final application is a realistic and interactive virtual museum which can successfully be operated with the Nintendo Wii controllers. The result of a subsequent usability test shows that it is highly usable as well.*

*The innovative input devices are very welcome by users in this kind of applications. The use of shadow techniques notably improves the visual appearance and sense of presence when exploring virtual environments. Even if there are still a few aspects which can be improved regarding the interaction, the overall result is very satisfactory.*



# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Background . . . . .	3
1.2	Aim of this thesis . . . . .	4
1.3	Outline . . . . .	4
1.4	Previous and related work . . . . .	5
1.4.1	Cultural heritage . . . . .	5
1.4.2	Virtual museums . . . . .	6
1.4.3	The Nintendo Wii controllers . . . . .	8
1.4.4	Interaction Techniques in virtual museums . . . . .	9
<b>2</b>	<b>An overview of virtual reality</b>	<b>11</b>
2.1	Virtual Reality . . . . .	11
2.1.1	Immersion . . . . .	11
2.1.2	Direct and implicit interaction . . . . .	12
2.1.3	Realistic modeling vs real-time rendering . . . . .	12
2.1.4	Application areas . . . . .	13
2.2	Interaction techniques and devices in virtual environments . . . . .	17
2.2.1	Input devices . . . . .	17
2.2.2	Output devices . . . . .	23
2.3	3D interaction techniques . . . . .	30
2.3.1	Pointing techniques . . . . .	30
2.3.2	Virtual Hand techniques . . . . .	30
2.3.3	World-in-miniature (WIM) . . . . .	31
2.3.4	Combined techniques . . . . .	31
2.3.5	Rotation techniques . . . . .	31
<b>3</b>	<b>Users and user requirements</b>	<b>33</b>
3.1	Context analysis . . . . .	33
3.2	Target groups . . . . .	35
3.3	Requirements . . . . .	35

<b>4</b>	<b>Modeling a virtual museum</b>	<b>37</b>
4.1	The environment . . . . .	37
4.2	Pictures and objects . . . . .	38
4.2.1	Scanning process . . . . .	40
4.2.2	Refining the models . . . . .	41
4.2.3	Object export and allocation . . . . .	42
<b>5</b>	<b>Interacting with the virtual environment</b>	<b>45</b>
5.1	EON Studio . . . . .	45
5.2	Navigation . . . . .	47
5.3	Interaction with virtual objects . . . . .	47
5.3.1	Selection . . . . .	47
5.3.2	Inspection mode . . . . .	48
5.3.3	The Wii controllers . . . . .	51
<b>6</b>	<b>Usability test</b>	<b>55</b>
6.1	Standards in usability and user-centered design . . . . .	55
6.2	Objectives . . . . .	56
6.3	Test group . . . . .	57
6.4	Testing method . . . . .	57
6.4.1	Hardware settings . . . . .	58
6.4.2	Tasks . . . . .	58
6.5	Results . . . . .	59
6.5.1	Questionnaire . . . . .	59
6.5.2	Interview . . . . .	63
6.5.3	Observations . . . . .	64
6.6	Discussion . . . . .	66
<b>7</b>	<b>Conclusions</b>	<b>69</b>
7.1	The museum . . . . .	69
7.2	Technical concerns . . . . .	71
<b>8</b>	<b>Future Work</b>	<b>73</b>
8.1	Selection . . . . .	73
8.2	Rotations . . . . .	73
8.3	Interface . . . . .	73
8.4	Exhibits . . . . .	74
8.5	Interaction . . . . .	74
8.6	Immersion . . . . .	74
<b>9</b>	<b>Acknowledgements</b>	<b>75</b>

<b>10</b>	<b>References</b>	<b>76</b>
<b>11</b>	<b>Appendices</b>	<b>83</b>
11.1	List of exhibits . . . . .	83
11.2	Code of the EON Studio Scripts . . . . .	85
11.2.1	InspectionMode . . . . .	85
11.2.2	ExhibitInfo . . . . .	97
11.3	GlovePie script . . . . .	100
11.4	Questionnaire and interview for the usability test . . . . .	101
11.4.1	Questionnaire . . . . .	101
11.4.2	Interview . . . . .	103





# Chapter 1

## Introduction

### 1.1 Background

During the latest years Virtual Reality (VR) has become more and more popular. We find one of its first and most influential antecedents in the flight simulator, already back in the years of World War II[1][2]. Soon it began to spread to many different fields such as the entertainment industry or industrial design, until the point that nowadays it is present almost everywhere. The reason for this are the multiple advantages it provides in comparison to real-life equivalent situations. These advantages include safety, time-related and economic reasons. Furthermore, Virtual Reality is not only a tool to extend the possibilities of already existing conditions; it also offers the possibility to create brand new virtual worlds.

One interesting application of this technology are the so called *virtual museums*. The term was coined 18 years ago by Tsichritzis and Gibbs when trying to define a *museum composed only by virtual artifacts, in a virtual setting accessible from a telecommunication network in a participatory manner*[3]. It is not only their contribution to cultural heritage what makes them so popular, but also their capability to make museums more attractive for the public of all ages. This can be achieved by complementing the exhibits with other multi-sensory information such as text, images, sound or video.

On the other hand, museums very often exhibit non-permanent collections. Once a temporary exposition finalizes, the exhibits are stored and not available for the public anymore. This is the concrete case of the Kulturen museum in Lund, which has many such exhibitions every year. For them it is not physically possible to have all the exhibits permanently exposed, but

it would be desirable to find a way to grant the public permanent access to them. The answer to their problem is a virtual version of all these collections, which in this way can be permanently exposed and even reachable from any place with Internet access.

## 1.2 Aim of this thesis

Based on previous work on interaction techniques in Virtual Reality museums, the purpose of this master thesis is to create a realistic and highly usable virtual environment (VE) for an existing exhibition.

## 1.3 Outline

To start we will introduce the concept of *Virtual Reality*, as it is the technology which all the thesis revolves around. Some devices and interaction techniques in Virtual Reality environments are also described in chapter 2.

In every application which is intended to be used by the public it is important to define a target user group. This is described in chapter 3, together with the application requirements.

We will move on with the bulk of the thesis, which can be divided in two big blocks. Chapter 4 corresponds to the modeling of the museum itself and the objects allocated inside. The navigation and interaction with the virtual environment are described next in chapter 5.

A study of usability has been carried out, so we can figure out the real applications of a project with these characteristics. The method and results of this study can be found in chapter 6.

Chapters 7 and 8 contain some personal conclusions and the future work that might be done.

To finish I would like to say a word of thanks to all the people who helped me out with this project. It can be found in chapter 9.

The final part of the thesis, which is composed by chapters 10 and 11, corresponds to the references and appendices respectively.

## 1.4 Previous and related work

Even if the field of Virtual Reality is not as established as other fields, there is still a lot of research done in this topic. We will try now to focus in some projects related to this thesis, such as previous work done in cultural heritage, virtual museums or applications which use the Nintendo Wii controllers as input device.

### 1.4.1 Cultural heritage

There are several projects developed with the intention to preserve the legacy of artifacts inherited from past generations, and also make all these pieces of art accessible to public all around the world. Before starting to develop this project we took a look at this work and would like to briefly introduce here some of these projects.

#### **Capturing content for virtual museums: from pieces to exhibits**

This project describes a novel method for digitally recording not only individual museum pieces, but also the rooms or spaces where they are exhibited. The detail of objects and scenes is very high (photographic quality), and it is possible to make measurements directly on the scene with millimeter precision. The methodology, its application to capturing museum exhibits, and examples of exhibits recorded using this technique are described in the paper[6]. This work is part of the Virseum project[7] at the School of Information and Library Science at the University of North Carolina at Chapel Hill (UNC).

#### **Modelling of 'Drottens' church ruins in a collaborative virtual environment**

The 'Drotten' church was active in the southern town of Lund, Sweden, for many hundreds of years. Nowadays its ruins are located under a bank in an underground museum which is closed to the public. A project developed at the RE-FLEX flexible reality centre in Lund, where this thesis has been performed, took care of creating an accurate representation of this church (see Figure 1.1). The virtual environments are still available online at the

RE-FLEX website[8], and also as a display at Kulturen museum in Lund[9]. The report describing this work can be also found online at [10](in swedish).



Figure 1.1: Virtual representation of the no longer existing Drotten church in Lund

## 1.4.2 Virtual museums

Virtual museums are directly related to the preservation of cultural heritage, but not all the work done in this field is related to the acquisition of virtual models of the artifacts. Next we will introduce a few projects which focus at different aspects of virtual museums, such as navigation or display control.

### **Tangible photorealistic virtual museum**

The aim of this project was the development of a photorealistic virtual museum that lets people interact naturally and have an immersive experience with exhibits. The system uses a technique that enhances a panorama with object-centered image sets, or object movies. A handheld 3D cube lets the visitor examine a particular artifact by using intuitive hand movements[11].

## **Adaptive Display Control of exhibits in a Distributed Haptic Museum**

An interesting adaptive display control scheme for touchable exhibits in a distributed haptic museum. Different levels of detail for the objects and proactive transfer control of the information files are used in order to improve the output quality of haptic, audio and visual media[12].

## **Real exhibitions in a virtual museum**

For some orientation concerning the creation of a virtual museum to be offered to visitors of real world museums one could take a look at this project's paper[13]. It gathers together different findings regarding not only the design of the application, but also the equipment used to run it. A number of design and implementation aspects are described along with an experiment designed to evaluate alternative approaches for implementing the navigation across the environment.

## **Design of a Walkthrough System for virtual museum based on Voronoi Diagram**

A new method of designing a walkthrough system for virtual museums. In the system, the empty 3D indoor scene of the museum can be projected in order to get a 2D polygon. This polygon and its Voronoi diagram are used to cooperative design and organize the scene contents, develop real time rendering and provide other interactive technologies such as visibility culling, collision detection or path planning. The system has been used in the archaeological Digital Museum of Shandong University (archaeological DMSDU)[14].

## **An XML-Based Scene Description Language for 3D virtual museum**

The purpose of this project was to create a language specially designed to describe virtual museum scenes. It is based on XML, with a simplicity that allows exhibits to be rapidly displayed on the Internet. Authoring tools and

a scene navigation system have been accordingly developed to display scenes described with this language[15].

### **Virtual museums for all - employing game technology for edutainment**

Finally, we present a case study based on an already developed version of a virtual museum and a newly implemented version that uses game technologies. The idea has the purpose of making virtual environments development be more affordable in terms of both development and hardware[16].

#### **1.4.3 The Nintendo Wii controllers**

As for Wii Remote -the main Nintendo Wii controller- projects refers, we must highlight the work done by Johnny Chung Lee, Human-Interaction PhD at Carnegie Mellon University. His projects in the field are probably the most original and well-known over the Internet. As examples we will cite next some of them. All the software can be found online at his website [17] and there are also several demo videos on YouTube[18]. An official discussion forum is available at [19].

Besides those developed by Lee we can find many other Wiimote projects online, such as a window curtain control and an application to set the color of a LED lighting[20]. The Wii controller offers endless possibilities and there is no lack of people interested in exploring them.

#### **Tracking your fingers with the Wiimote**

The infrared camera in the Wii remote is used to track our own your fingers in 2D space. This lets the user interact with a computer by waving the hands in the air, similar to the interaction seen in the movie "Minority Report".

#### **Head Tracking for Desktop VR Displays using the Wii Remote**

This time the infrared camera is combined with a head mounted sensor bar to track the location of the user's head and render view dependent images on

the screen. The display properly reacts to head and body movement creating a realistic illusion of depth and space, as if we were looking through a window to a virtual environment.

### **Low-cost Multi-point Interactive Whiteboards using the Wiimote**

This application permits the tracking of pens that have an IR led in the tip. By pointing a Wiimote at a projection screen or LCD display, very low-cost interactive whiteboards or tablet displays can be created.

#### **1.4.4 Interaction Techniques in virtual museums**

Last but not least we introduce the work done by Dirk Verhagen in interaction techniques in virtual museums[4], a master thesis developed at the Dept. of Design Sciences at Lund University where the current thesis has been performed[21]. His work focused on the comparison of a number of different interaction techniques that can be used in a Virtual Reality museum framework. The results of this study, which states that Nintendo Wii controllers are the most recommended within the set of interaction devices tested, have been used as basis for this thesis.





# Chapter 2

## An overview of virtual reality

### 2.1 Virtual Reality

Virtual Reality is a technology which allows a user to interact with a computer-simulated environment. Every application based on it should count on four basic factors: a digital model, real-time navigation, immersion and direct interaction. The result of these elements mixed together is a sense of presence that is capable to produce reactions in the user as if the experience was real. In the end most of the so called virtual environments are primarily visual experiences, but there are many different ways of interaction that provide the user with different levels of reality and affect the senses in different ways[22].

#### 2.1.1 Immersion

Slater and Wilbur (1997) define immersion as a description of *the extent to which the computer displays are capable of delivering an inclusive, extensive, surrounding, and vivid illusion to the senses of a human participant*. [23]

According to this definition a system is immersing when it disconnects the senses from the real world and connects them to the virtual world. Sensory and 3D immersion means that Virtual Reality intends to substitute our perception, bound to the real world, for another perception generated by computer.

Although this immersion can involve different senses, the most important is

the sight. That is why in most VE's the efforts are focused on the appearance of the world.

The acoustic immersion is the easiest one to achieve, but its usage is not so popular because it avoids the communication between participants in collective experiences. Nevertheless in a single user application we can construct a quite immersing environment by providing only visual and acoustic information.

Another important sense in VR is the touch. When it is involved we talk about haptic immersion. Although the word *haptic* comes from the Greek word *haptesthai*, which means "to touch"[24], haptic devices provide both force and tactile feedback. The latest advances in Virtual Reality are oriented to complete the experiences with tactile sensations as well.

Other senses such as the vestibular sense -which contributes to our balance and our sense of spatial orientation-, smell or taste are hard to immerse, so it is quite difficult to find a VR application which provides this kind of feedback.

### **2.1.2 Direct and implicit interaction**

In this new paradigm of interaction we leave behind the old-fashioned WIMP interfaces (Windows, Icons, Menus and Pointers) to give way to a more natural usage of the applications. The user does not have to say what he/she wants to do anymore but only needs to act in a natural way. We talk about *direct interaction*. The typical mouse is replaced by sensors and a gestural interface which interprets our movements and can modify the perspective projection according to our position and the orientation where we are looking at. As the computer knows exactly where we are without having to do anything explicitly we are talking about *implicit interaction*.

### **2.1.3 Realistic modeling vs real-time rendering**

As has been discussed previously, achieving a visual immersion is the most important in a realistic VE. Therefore we could think high resolution and detailed models are always welcomed in this kind of applications, but it is not so easy when we have to deal with real-time navigation at the same time.

Real-time navigation refers to the possibility of moving at will through the virtual environment and exploring it as we wish. There must be a short

”cognitive distance” between a user’s action and the system’s feedback that shows the result of that action, but manipulating and rendering a high-detailed 3D model requires calculations that cannot be done so fast. We need to find a balance here and minimize this time of calculation while keeping the realism of the models as much as possible.

The improvements in the design of graphic cards and new shader techniques allow us to create increasingly real virtual environments, but it is still early to know what we are going to achieve during the next years.

## 2.1.4 Application areas

When talking about Virtual Reality we are referring basically to *simulation*.

**“Simulation - the act of imitating the behavior of some situation or some process by means of something suitably analogous” [25]**

Even if Virtual Reality is generally associated with leisure and entertainment applications, its usage is increasing more and more in many different fields such as industrial design or training.

### Design

The benefits of applying Virtual Reality to industrial design are numerous[26]. Let’s imagine we need to design a new car. The cost of designing it, building all the pieces and then assembling them is large, and the process is not only expensive but also time consuming. Nevertheless, by constructing a virtual model of the car we can skip all the middle steps and go directly from the design to the testing, which is way faster and cheaper, and build it only once we are satisfied with the results.

Now let’s also imagine we work for a multinational company which has headquarters in different parts of the world. In this kind of companies it is quite common to design the products in one country and do the testing in another one, so very often we will have to share our work with people who is many miles away. Again it is much easier, faster and cheaper to share a digital model via mail in a DVD, or even better via Internet, than to send the real product or mobilize the people.

## **Education and research**

For educational purposes VR can be used to facilitate learning, helping students to understand complex systems and processes. It is an useful tool in long distance learning as well[27][28].

Since virtual laboratories can be built instead of real ones at a fraction of the cost, it is also a solution to extend school boundaries in an economical way. Moreover this can be done by using the same computers schools already have.[29]

We can find ways to apply VR in many different research fields as well. In medical research, for example, we can find VR applications for the study of human behavior under conditions of stress[30], or for the analysis of genomic data[31]. It is also a very resorted tool in physics simulation. In the archaeological field this kind of applications contribute to the preservation of cultural heritage, and the data can be also used to further education and analysis[22].

## **Training**

Medical and military training are two fields that entail a great deal of risk and responsibility. Hence VR applications for surgery training and battlefield simulation are becoming more and more common. Now it is possible to model a heart with problems and prepare the surgical intervention without danger, by means of virtual interaction with the organ to operate. In the military field computers are used to simulate military situations and conditions, usually for the purpose of developing and testing operational plans or training. Flight simulation -which was actually one of the most influential antecedents of VR- and maintenance training are also common applications in this area.[32]

## **Leisure**

As for leisure and entertainment refers the videogame industry is probably the most worth mentioning. Just like the military and industry, it has been the source of many valuable spin-offs in VR. Yet in the 80's we find videogames such as Battlezone[33] (see Figure 2.1), which is widely considered the first Virtual Reality arcade game because of its use of first-person 3D graphics, combined with an actual "viewing goggle" that the player put his face into.

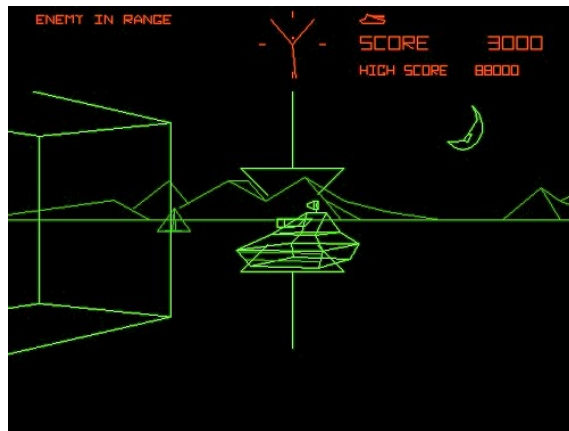


Figure 2.1: Battlezone

In the 90's a pioneer group in the field of Virtual Reality Entertainment appeared. It was called *Virtuality* and delivered the first VR public entertainment machines which provided real time, stereoscopic and networked interactive VR experiences. The gaming system called the 1000CS[34], released in 1991, was a stand-up immersive head-mounted display (see page 24) platform with a tracked 3D joystick.

Some years after that, in 1997, the well-known companies Sony and Nintendo revolutionized the videogame market with the first rumble controllers, which were probably the first approach to haptic devices in domestic game consoles. Nowadays we can find a wide variety of controllers such as guitars, tennis rackets or steering wheels, and some others are starting to include movement sensors. We can also notice the evolution of the industry in the quality of the graphics, which are becoming more and more real as time goes by.

But although the videogame industry is a natural consumer of computer graphics, there are many other applications besides gaming. We can find it in art, cinema or even have a parallel virtual life at the Internet with applications such as Second Life[35] (see Figure 2.2).

Virtual cinematography is an emerging field that consists basically in cinematographic techniques performed in a computer graphics environment. The technique involves shooting real people and scenes and then manipulating them digitally. A virtual camera is used to create viewpoints automatically, and then they can be integrated into a film or videogame. It is the same process previously used in computer-animated films, but using real people[36][37][38][39].

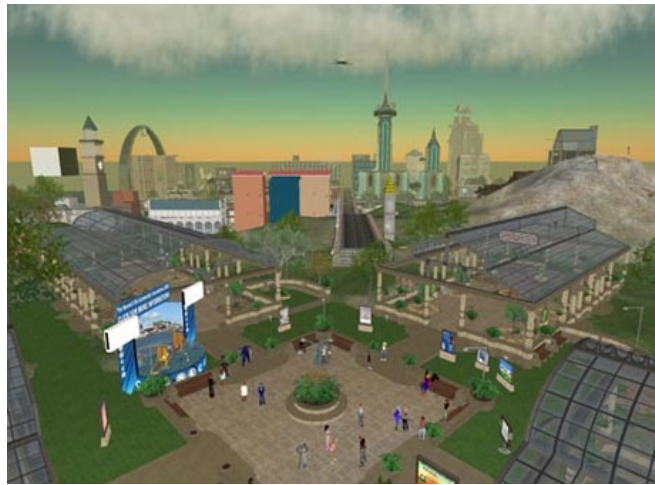


Figure 2.2: Second Life

Virtual museums are also an interesting application of VR. Besides their contribution to cultural heritage, they provide traditional museums with new features and allow the visitor to have a more interactive experience. An example of this kind of museums is shown next in Figure 2.3).



Figure 2.3: Example of virtual museum

## 2.2 Interaction techniques and devices in virtual environments

At this point we know that the aim of a VR application is to provide the user with a convincing sense of presence, to make the senses involved in the experience as much as possible, to make it seem real. It is obvious that the input and output devices used to interact with the VE are an important part of this kind of applications, since once a user is immersed he/she should be capable to interact with the environment using natural skills. As there are many different kinds of IO devices we will describe here the most commonly used in VR[40][41].

### 2.2.1 Input devices

#### Standard input devices

This section covers the most basic input devices, those that we are used to use in our daily life. We can also refer to them as *desktop devices*.

- **Keyboards** are a traditional device consisting of buttons which can be used to either input a linguistic character to a computer, or to call upon a particular function. Keyboards are commonly used as input devices in some applications such as computer games, but they are not practical when bringing them into more immersive environments since in most of them the users are standing. Nevertheless, entering alphanumeric information is important in many 3D applications, and less conventional variations can be used instead of the traditional keyboard.

A *keyset* or *chorded keyboard* (see Figure 2.4) allows the user to enter characters formed by pressing several keys together, like playing a "chord" on a piano. Due to the large number of combinations available from just a few keys it can be used with one only hand. A chorded keyboard minus the board, typically designed to be used while held in the hand, is called a *keyer*.

- **Mice** are the most used pointing devices in traditional 2D applications. Their two essential components are a 2D locator generating relative 2D coordinates, and a set of buttons (usually 2 or 3 and a small wheel to scroll up/down). Mice are often combined with



keyboards in computer games to provide more interaction possibilities, but since they need to be placed in a 2D surface they are not suited for more immersive environments.

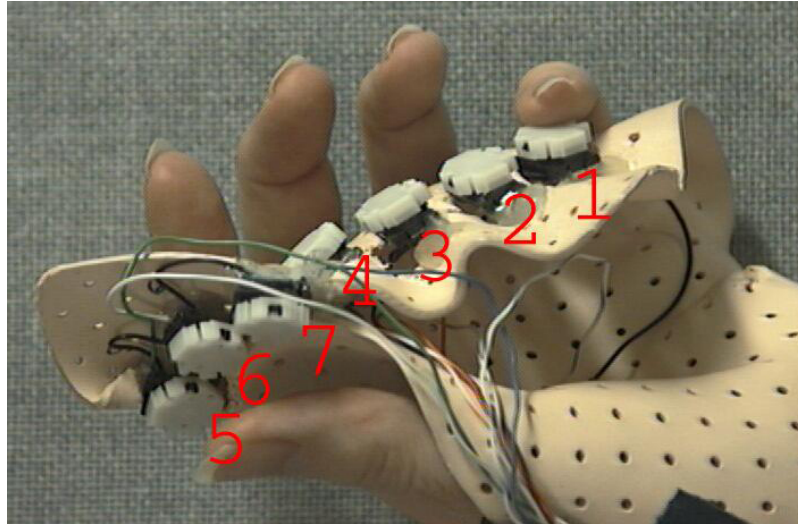


Figure 2.4: Keyer

- **Trackballs** are basically upside-down mice. The difference lies in moving the ball embedded in the device instead of moving the device itself, which is an important advantage as they do not need to be placed in a surface to work. The fact that a trackball can be held in one hand makes it more likely to be used in an immersive 3D environment, but the user still needs the other hand to manipulate the ball which is a big inconvenience. In most cases 3D mice (see Section 2.2.1) are used instead.
- **Joysticks** , as mice and trackballs, have a continuous 2D locator and a set of buttons, and it has been commonly used on the desktop even if it was originally used to control aircraft's ailerons and elevators. The main difference is a stick that pivots on a base and reports its angle or direction. Also, instead of stopping moving as soon as the device stops moving (as mice and trackballs do), the cursor typically continues moving in the direction it points to until it is returned to the neutral position. Joysticks are frequently used in driving and flight simulation games and other desktop and console applications. Even if we can find handheld versions of them, they are rarely used in 3D applications that employ HMDs (see Section 2.2.2) or surround-screen visual displays (see Section 2.2.2).

- **Six-DOF desktop input devices** This kind of devices have been specifically developed for 3D interaction on the desktop, and are commonly used to manipulate virtual objects in combination with the mouse. They allow translation and rotation about three perpendicular axes (move forward/backward, up/down, left/right, roll, yaw, and pitch) while menu items can be selected with the mouse. Since the devices work better when grounded they are usually used in CAD/CAM and animation applications, but rarely in more immersive environments. Another downside is that they are hard to use[41][43].
- **Touch screens** are displays which can detect the presence and location of a touch within the display area. The interaction is done with the fingers or by using a passive object (i.e. an stylus), directly on the image shown on the screen. The most important is that no intermediate devices are needed. Thanks to the multi-touch technology[44] these screens are capable of sensing several points of contact at a time, and can also detect how hard we are pressing on the screen. In the latest years this kind of screens have been used in many devices such as digital cameras, mobile phones, PDAs and even portable consoles (see Figure 2.5), as a way to interact with menus and also to play videogames[41][45].



Figure 2.5: Portable console Nintendo DS

### 3D mice

Tracking devices (see Section 2.2.1) are usually combined with other components such as buttons, sliders, knobs and dials. These components are called 3D mice, as seen in Figure 2.6. We differentiate them from traditional 2D mice because the user physically moves them in 3D space to obtain position and/or orientation. Therefore, users can hold them or even wear them, and in addition they can be mapped to many different interaction techniques.



Figure 2.6: 3D mouse

### Tracking devices

Trackers detect head, hand or body movements and send that information to the computer. The computer then translates it and ensures that position and orientation are reflected accurately in the virtual world. Tracking is important in presenting the correct viewpoint, coordinating the spatial and sound information presented to users as well the tasks or functions that they could perform.

- **Motion tracking** is fundamental in many interaction techniques, as most of the times we need to have a correspondence between the physical and the virtual world. There are several types of 3D motion trackers: mechanical, magnetic, acoustic (or ultrasonic), optical, inertial and hybrid.

**Mechanical trackers** consist of one fixed and one mobile end which is attached to the object to track. Both ends are interconnected with mechanical linkages, and several electrome-

chanical transducers (potentiometers, shaft encoders) take position and orientation measurements as the tracked object moves.

**Magnetic trackers** use a transmitting device that emits a low-frequency magnetic field. A small sensor receives the signal and uses it to determine its position and orientation relative to the transmitter. Since its range is limited they are quite accurate, but not appropriate for large display environments. Moreover, any metal object located in the same room can cause distortions in the magnetic field, and thereby reducing the accuracy.

**Acoustic trackers** work in an analog way, replacing the magnetic field by high-frequency sound, and the sensor by microphones. The distance between emitter and receiver is determined by the time it takes for an ultrasonic pulse to travel from the source to the destination. Just like magnetic trackers they work only in short ranges and can be easily distorted, but their weight is lighter and their price cheaper.

**Optical trackers** determine position and orientation from measurements of reflected or emitted light. Cameras, infrared emitters and other optical sensors are used, and in this case the possible accuracy problems is caused by occlusion.

**Inertial trackers** were originally used in ships, submarines and airplanes during the 50's. Different inertial measurement devices, such as gyroscopes and accelerometers, are integrated to provide angular velocity and linear acceleration information. Since the tracking system is in the sensor, their range is limited either by the length of the wire which connects them to the electronics unit, or by the range of the wireless system. The major disadvantage is the error accumulation from sensor biases, noise and drift.

**Hybrid tracking systems**, finally, combine several tracking technologies to increase accuracy and reduce latency.

- **Eye tracking** These devices track the movement of the eyes by capturing the corneal reflections with a camera. The objective is to determine where the user is looking, for example for psychological evaluation or as a tool to interact with an application. These devices can be worn or embedded into a screen, and can be very useful to improve the accuracy of gaze-directed techniques which

use head-trackers as an approximation of where the user is looking. Figure 2.7 shows an example of eye tracking device.



Figure 2.7: Eye tracking device

- **Data gloves** provide information about the movement of the user's hands and fingers. We can distinguish two varieties:

**Bend-sensing gloves** use light-based, fiber-optic, resistive ink and strain-gauge bend sensors to provide, basically, angle measurements in order to detect hand postures. Typically between 5 and 22 sensors are used, depending on the accuracy needed. These kind of gloves provide a large number of degrees of freedom (DOF) but they have to be worn by the user, and sometimes they need calibration to suit the user. An example can be seen in Figure 2.8.

**Pinch globes** generate an electrical signal every time the user touches two or more fingertips together. They are used to detect grabbing and pinching gestures in the context of object selection, mode switching and other techniques. Their extreme lightness, multiple input combinations and two-handed interaction are important advantages of these devices. A prolonged use, however, can make them unusable.

Both varieties can also be combined in one glove to achieve better accuracy. This is the case of the *Flex and Pinch* prototype.



Figure 2.8: A bend-sensing data glove

## 2.2.2 Output devices

### Monitors

Conventional monitors are widely used in many different kinds of 3D applications. They provide monocular and motion parallax effects, and also stereopsis (depth sense, 3D effect) when combined with stereo glasses and a capable graphics card.

### Powerwalls

The Powerwall is a large (from 2 to 20 meters and even more) high-resolution display wall that shows images composed by several smaller high resolution images. This makes possible to maintain image fidelity even at very close viewing distances. An example of use is industrial design: we can design a new car model and render it on the Powerwall in real size to analyze its quality[41][46], as shown in Figure 2.9.



Figure 2.9: A powerwall used for automotive design

## Head-mounted displays

HMDs are visual displays attached to the user's head (see Figure 2.10). The images are placed right in front of the eyes in one or two small screens, for monoscopic and stereoscopic viewing respectively. A combination of lenses and/or mirrors are used to show and sometimes magnify the images on the screens, which can be either LCD or CRT. Also, sometimes a camera is mounted on the device allowing it to display both real and virtual objects.

An alternative to this are the so-called head-mounted projective displays (HMPD), which combine HMDs and projective displays. Small LCD projectors project the images into the real environment, and a special retroreflective material is placed strategically in the environment so that the user sees the graphics reflecting off it.



Figure 2.10: Head-mounted display

HMDs provide the user with complete visual immersion, but due to their small field of view (FOV) some distortions in perception of size and distance might appear. Some users can also have a problematic stereo viewing due to the variation in the different interocular distances, and the device may not fit every user because of the different head sizes and shapes. Another disadvantage is that the viewers cannot see each other in multiple user experiences. Some advantages are their portability and lower cost when comparing

them to projection-based displays. In multiple user experiences every user can have his own tracked view of the virtual world, even if the high cost of the hardware needed can be prohibitive as multiple HMDs (and therefore multiple graphic engines) are required[41][47].

### **Surround-Screen displays: CAVE System**

Developed at University of Illinois, Chicago, the Cave Automatic Virtual Environment -most commonly known as CAVE- is an immersive multiuser interface for VR applications. Projection screens are placed on the walls, floor and/or ceiling of a "room" (see Figure 2.11) where the images of the VE are projected. The users are placed in the middle of this room wearing a tracking device and special glasses. The computer generates one different image for each eye at a frequency that is twice the one used in standard projections, and the glasses are synchronized with it by IR so that each eye only sees its corresponding image. The brain fuses both images into one so that it seems to the user like if he was seeing 3D objects. This way of simulating relief images is called *stereoscopic projections*.

Since the system cannot manage the tracking of more than one user there is only one of them wearing the tracking system. The images are generated in the right perspective according to the position of this "guide" user. The others can still see the images with the special glasses but with a less realistic effect, since they are not placed in the optimal position which is the one determined by the user.

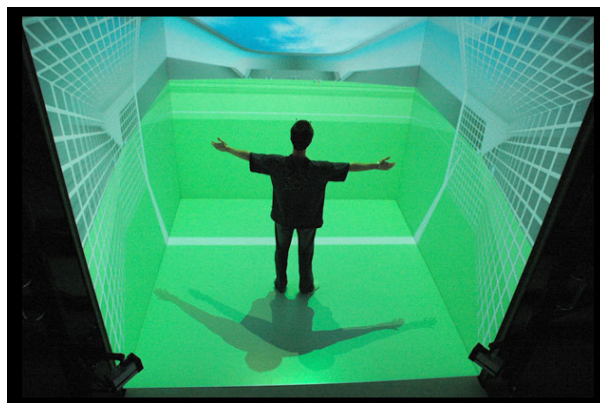


Figure 2.11: 4-walled CAVE



The audio system is also configured in such a way that it gives a 3D sense, and the position sensor is again used as reference to determine from which direction the sound comes and at what distance the user is from the sound source.

The walls are retro-projector screens, which means the projectors are positioned outside the room and the floor is projected from the ceiling (frontal projection). To reduce the size of the whole system the images are not projected on the screens directly. Instead they are projected on special mirrors so that the projectors can be placed closer to the screens.

CAVE systems can come in many different configurations, depending on the number and type of screens they have:

- 2-walled (front and floor / front-left and front-right)
- 3-walled (front-left, front-right, and floor or ceiling)
- 4-walled (left, right, front, and floor or ceiling)
- 6-walled (all 6 sides of the cube)

Two different kinds of interaction take place in these environments: the modification of the VE, which depends on the position and orientation of the guide user as said before, and the movement inside the VE, which is done by means of a special joystick also carried by the guide user. This joystick has his own position sensor and allows the user to interact with the environment and also to move across it without having to move physically. The rest of functionalities depend on the application[41][48].

## **Haptic devices**

As said previously, haptics refers to a technology which allows a user to get tactile and force feedback from an application, such as the texture of surfaces, vibration, temperature or weight. As they are input-output devices the interfaces not only look but also feel when interacting with them, unlike most common computer interface devices where the information flows only in the human-computer direction[49][50].

The usage of this kind of devices can bring benefits to any tasks requiring hand-eye coordination, such as force feedback in gaming (feel the impact of

the ball when playing a tennis game) or safe training in medical environments (surgery).

Haptic display devices can be divided in five categories[41]:

- **Ground-referenced haptic devices**, also called world-grounded devices, can be grounded to any surface (which means their range is often limited). A physical link is created between the user and a point in the ground to provide force-reflecting and/or force-feedback. Electric, pneumatic or hydraulic actuator technology are typically used. Some examples of this kind of devices are force-reflecting joysticks (see Figure 2.13), pen-based force-feedback devices (as the one showed in Image 2.12), motion platforms or large articulated robotic arms.



Figure 2.12: Pen-based force-feedback ground-referenced device located at the Virtual Reality lab, IKDC, Lunds Universitet



(a) The Falcon, developed by Novint, is a commercial force-reflecting joystick[51]

(b) PHANTOM, another example of pen-based force-feedback device[52]

Figure 2.13: Force-reflecting joysticks

- **Body-referenced haptic devices** can be grounded to the user in some way. The freedom of motion is consequently higher, but as the user has to wear the device there are some weight and size limitations. This type of device usually uses electrical or pneumatic actuator technology and can come in two varieties: hand-force-feedback devices and arm exoskeletons.

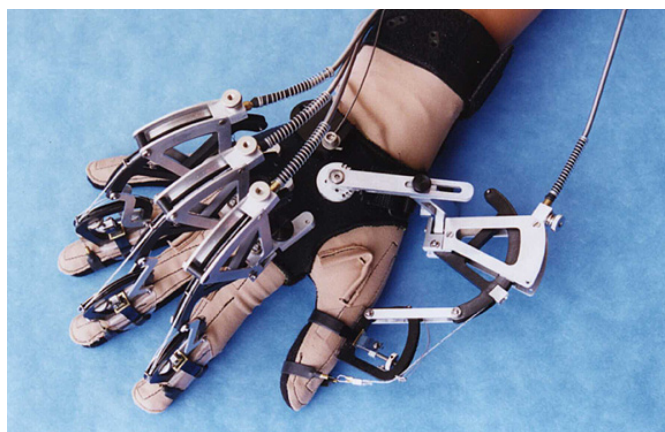


Figure 2.14: Hand-force-feedback glove

- **Tactile devices** stimulate the user's tactile sense by using vibrators (i.e, this is the case of some console gamepads, as seen in Figure 2.15), electrical currents, temperature-regulating devices and other actuators. There is also another kind of devices that are placed behind the user's ear in order to stimulate the vestibular sense, which can create a sensation of motion.



Figure 2.15: The XBox 360 gamepad includes a rumble feature

- **Passive haptic devices** provides a constant force or tactile sensation based on the geometry and texture of the particular object. The shape of the physical object matches the one of the virtual object, so it provides a fully realistic haptic sensation. This approach is less flexible than the real force-feedback devices since the device used is very specific. An example of passive haptic device is shown in Figure 2.16

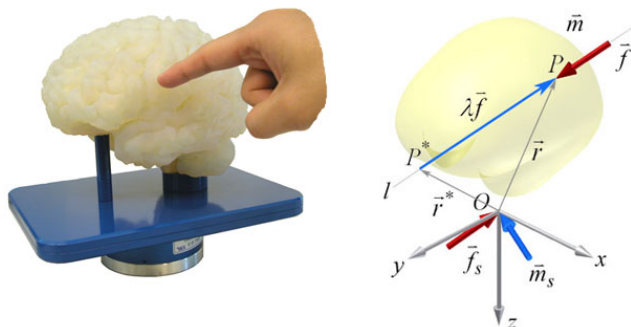


Figure 2.16: Passive haptic device and principle of operation

- **Combination of devices** A combination of different types of devices to achieve a stronger haptic sensations.

## 2.3 3D interaction techniques

Interaction techniques are the bridge that connects software and hardware components in 3D interfaces, the methods that allow us to manipulate the objects in the virtual world without modifying their shape. The following are the most basic manipulation tasks[41]:

- **Selection** involves picking one object in particular amongst a set
- **Positioning** is the task of changing the object coordinates in the virtual world
- **Rotation** changes the orientation of an object

Different interaction techniques can be applied to these tasks.

### 2.3.1 Pointing techniques

The techniques belonging to this group allow the user to select and manipulate the objects by simply pointing to them. The object selected is typically the one which is intersected first in the pointing direction. Some of these methods are the Ray-Casting, Flashlight and Image-Plane techniques.

### 2.3.2 Virtual Hand techniques

These techniques allow the user to manipulate the virtual objects with her hands, by mapping the position and orientation of the input device onto the ones of a virtual hand. Simple Virtual Hand and Go-Go are some examples of these methods.

### 2.3.3 World-in-miniature (WIM)

A small scale model of the virtual world is used so that the user can directly manipulate the objects placed in it.

### 2.3.4 Combined techniques

Since not all the techniques previously mentioned can be applied to any scenario or manipulation situation, the combination of different methods seems to be a good alternative. In *aggregation of techniques* methods the user can choose which of these methods prefers amongst a given list. On the other hand, in the *technique integration* approach the interface switches transparently between them depending on the manipulation task and context. HOMER, Scaled-World Grab and Voodoo Dolls are some of the techniques included in this group.

### 2.3.5 Rotation techniques

These kind of techniques can be used to slow down or accelerate the rotation of the user device, as typically these rotations are directly mapped to the virtual object. In the first case we can achieve better control of the rotations, which can be useful in applications such as teleoperated surgery. The advantage of using acceleration methods is that large rotations can be done with small rotations of the input device. This can be useful when the manipulated object is big, or when the user has movement restrictions.



# Chapter 3

## Users and user requirements

The final product of this thesis is an application likely to be used in a real museum. Therefore, we have to keep the end-user in mind while designing it, as well as the requirements of the Kulturen itself. This chapter tries to explain which those requirements are, and justifies the different design decisions that have been made.

### 3.1 Context analysis

Our exhibition is composed by 23 pieces (13 objects and 10 pictures) representing different aspects of Lund's history, classified as follows:

- Lund becomes Swedish, the city during the 17th and 18th centuries (3 pictures)
- The University (1 object + 2 pictures)
- The inhabitants of Lund (1 object + 2 pictures)
- Small-scale industry in Lund (8 objects + 1 picture)
- Medical history
- When the railroad came to Lund (2 objects + 1 picture)
- Second World War (1 object + 1 picture)



Due to the relatively small size of the exhibition a quite simple environment has been accordingly designed. Nevertheless, we decided on some desirable characteristics that have been taken in consideration:

1. Physical classification of the exhibits, since they pertain to different categories
2. Comfort while walking around the museum and inspecting the exhibits
3. A pleasant visual experience

To satisfy the first requirement we thought about having different corridors. One corridor per section seemed to be a good idea, but the fact that some of the groups have only 1 or 2 exhibits made us decide to group some of the sections. Having too many corridors does not help satisfying the second and third requirements either.

In order to make the museum comfortable and pleasant to the eye, the pictures and objects have been distributed in a balanced way (see Figure 3.1 for detailed information about the distribution). It is also preferable not to have too long or too short corridors, and have easy and quick access to the exhibits. Having in consideration all the previous arguments the number of corridors has been reduced to 3, following a star distribution. Also with the intention to improve the visual aspect and avoid having a sense of claustrophobia while walking around, a round, vaulted hall has been placed where the corridors meet. The hall makes the museum more spacious and also constitutes a good starting point for the simulation.

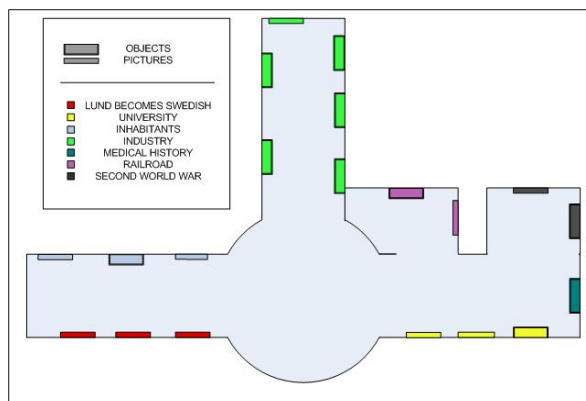


Figure 3.1: Distribution of the exhibits by sections

Both walls and floors have been covered with realistic textures, and lights and shadows have been used to make it feel like a real environment.

## 3.2 Target groups

Defining a target group for a museum is a challenge. It depends on which kind of objects we want to exhibit (*what*), and in which way we are going to show the information related to them (*how*). Even so, we cannot always predict which kind of people might be interested in it.

In our concrete case we have a predefined *what*: a collection of objects belonging to the 17th and 18th centuries' history of Lund. The subject is likely to be interesting for all sorts of people, so we will assume that our application should meet the expectations of both kids and adult visitors. Having an established target group will help us design the *how*, that is, designing the user interface and decide which information to show.

We will describe three groups of users:

**Group 1: Kids and teenagers** Below 18 years old. Not so interested in reading lots of text, more interested in playing and trying new things instead. Learn easily.

**Group 2: Young and middle-aged adults** 18-50 years old. Probably already used to technology, therefore more capable but also more demanding.

**Group 3: Advanced-aged adults** Over 50 years old. Not so used to technology, or not used to it at all. Some of them may have movement restrictions. Learn slowly.

## 3.3 Requirements

Considering the wide variety of users that we have and the specific needs of a museum, the following user requirements have been established:

- High usability: easy-to-use graphical interface and easy-to-learn interaction device

- Presentation of the content: attractive content and interaction possibilities for all ages
- Realism with regards to visual appearance and navigation
- Good response time
- Error tolerance

Keeping in mind these requirements the following decisions have been made:

**High usability** Based on previous work on interaction techniques in virtual reality museums[4], the Wii Remote and Nunchuck controllers have been chosen as interaction devices. The mapping of the controls follows the Nintendo policy, which makes it easier for users already familiar with Nintendo Wii. The controls used in navigation follow the standards commonly used by other applications. The user interface has been designed to be user-friendly, as simple and intuitive as possible.

**Realism** Since we are attempting to replace a real exhibition with a virtual one, we may want to keep on providing the visitor with real and comprehensive information. Thus, we should be as loyal toward reality as possible when representing them in the virtual world. To satisfy this requirement 3D scanning and texturizing techniques have been used. This process will be widely described further in this document. To give realism to the navigation the engine has been provided with collision detection, and a first person view has been used instead of a free camera.

**Presentation of the content** Besides the graphical and written information, other multimedia content (i.e. sound samples) has been added to stir children's interest. The content is showed in a clear and pleasant way.

**Response time and error tolerance** The size of the meshes has been reduced to improve the time of rendering. Also multiple levels of detail are used to accelerate the textures loading time. EON Studio has been used as tool to develop the navigation and interaction framework. The use of a commercial application guarantees a minimum tolerance to errors.

# Chapter 4

## Modeling a virtual museum

### 4.1 The environment

The first step to build the museum was to create the environment which would contain the exhibits. The modeling has been done with Autodesk 3DStudio Max 9 and Adobe Photoshop CS has been used to create the textures.

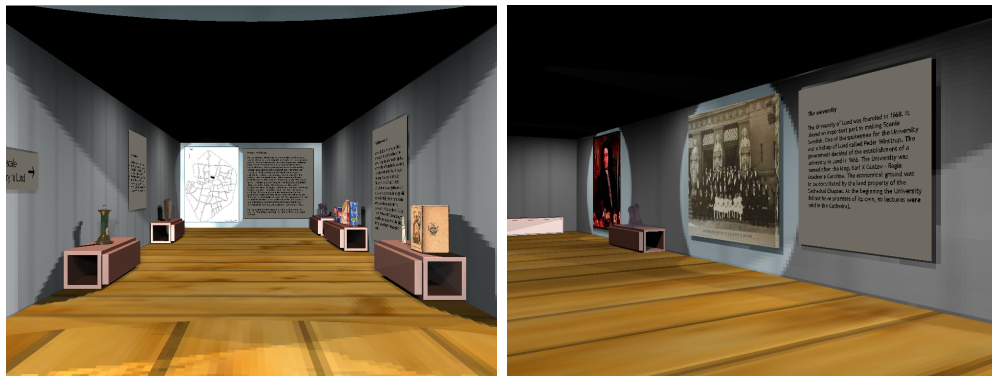


Figure 4.1: Some views of the museum

Some spotlights have been placed in strategic points so that the generated shadows increase the realism of the scene, and also make the visitors direct their attention on the exhibits. To accelerate the rendering speed we used render to texture instead of real-time generated shadows. Finally, some plants and furniture have been added as decoration (see Figure 4.2).

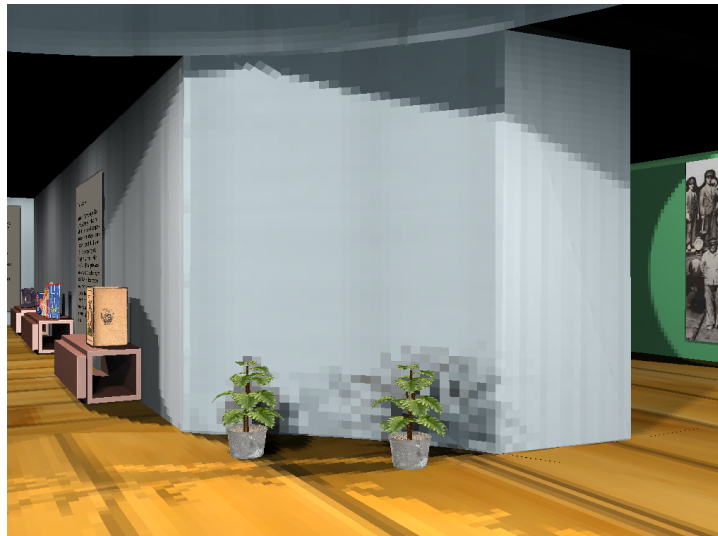


Figure 4.2: Picture showing some decoration and shadows generated with render to texture

For further implementation of the navigation engine we need to export the model to an EON Studio file (.eoz). The best way to do so is by using EON Raptor, a 3D Studio Max plugin conceived specially for this purpose and available for a number of 3DStudio Max versions.

## 4.2 Pictures and objects

As has been previously mentioned we distinguish between two kinds of exhibits -objects and pictures- with different modeling processes. The complete list of pieces can be found in the Appendices in Section 11.1

All the pictures have been created as simple boxes with a texture on them with 3D Studio Max and Adobe Photoshop. To keep a good level of detail the textures have been applied at a high resolution. The same process has been followed to model some simple objects such as tobacco or food packages.

The modeling of the objects with a more complex shape, however, requires several steps. The whole process is described next.



(a) Phone



(b) II World War bomb



(c) Mouthpiece for tobacco smoking



(d) Wrong burnt pot



(e) Slipper from an inhabitant in Nöden



(f) Food and tobacco packages



(g) Piece of railway

Figure 4.3: Some of the objects shown in the museum

### 4.2.1 Scanning process

In order to get an initial base mesh for the objects they have been subjected to a scanning process. The Next Engine Desktop 3D Scanner has been used to do so, and the Next Engine Scan Studio as processing application.

As shown in Figure 4.4 the objects are placed right in front of the scanner, which captures the data with multi-laser technology. One advantage of this hardware is that the textures are simultaneously generated and assigned to the mesh. Typically we will have to try different settings depending on the shininess, size and shape of the object to obtain a quality scan.



Figure 4.4: Scanning the objects

Once the scanning process finishes we will have several independent meshes, which correspond to the data captured from different points of view. Probably we will have to crop them to remove non-desired parts of the mesh, and also fill the places where no data could be captured by the scanner.

Next we proceed with the alignment of the meshes (see Figure 4.5), which is one of the most important steps. The Next Engine Scan Studio allows us to do so by placing some pins in different points of the meshes. Once we are satisfied with the result we can use several modifiers to improve the quality of the mesh. An useful tool is the one called *fuse*, which performs a volume merge, a remesh, intelligent simplification, and optionally fills holes. The resulting mesh is usually not that good and another refinement step is needed. We consider the tools provided by this software just as a first basic refinement.

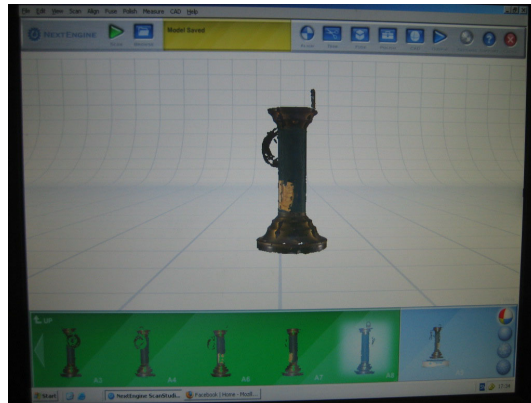


Figure 4.5: Aligning the captions

### 4.2.2 Refining the models

The refining of the meshes involves achieving a better alignment, filling the possible holes and, sometimes, applying some modifications to the texture information. This process has been done by using 3DStudio Max 9/2009. To switch from the Next Engine Scan Studio to this other application we need to export the meshes to a format readable by 3DS Max, like .obj, and then import them from 3DS Max. Although a great amount of the thesis' time has been invested in this process, modeling is not the main focus of this thesis and we will not give further details.

The following pictures show one of the meshes before and after its refinement.

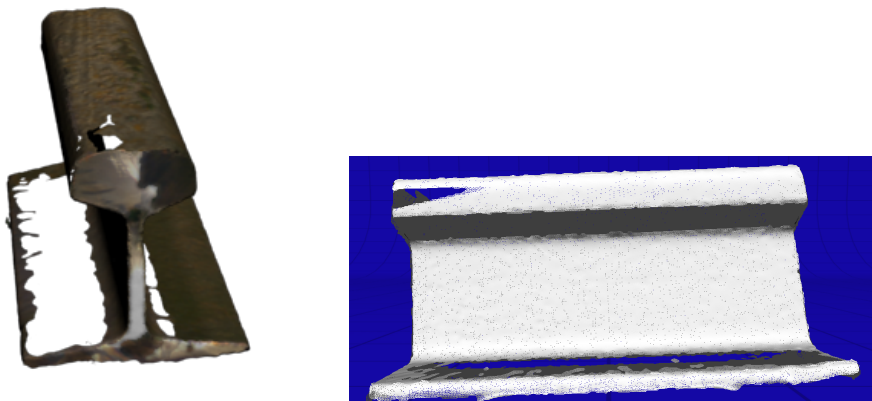


Figure 4.6: Mesh obtained right after the scanning process



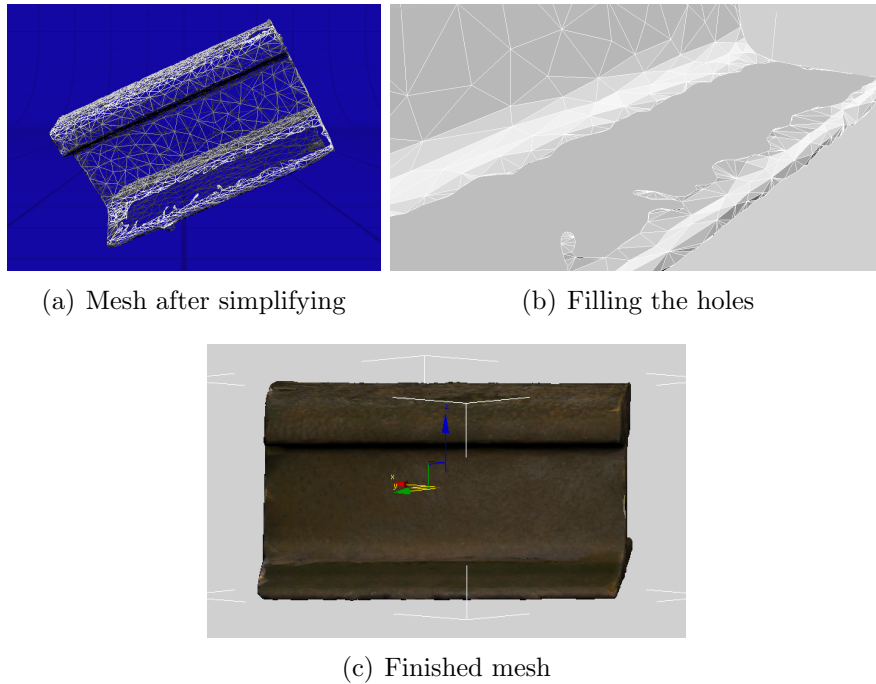


Figure 4.7: Refinement process

### 4.2.3 Object export and allocation

The next step is to allocate the objects inside the museum. Either we can export the .max files containing the models to .3ds files, which can be imported later on EON Studio, or use Raptor to generate a new .eoz file per object. In the latter case we will need to copy the converted textures and meshes to the .eoz file containing the museum. After trying both systems it seemed Raptor worked better, therefore this is the one we used.

Nevertheless, the process was not that easy, even though it seems all the tools are provided by EON. The reality was that we had to face many problems regarding the exporting-importing process due to the instability of Raptor, which sometimes was crashing and some other times could not export the textures properly. In the end, in most cases the only solution was to create the textures in EON Studio one by one and assign them to the meshes, a process that was too time-consuming since every object was using 10 different textures at least, and also there was no way to guess in which order we should assign them. As no alternative solution was found, we looked for a way to simplify the process by reducing the number of textures used per object.

We introduce here another modeling tool called Blender, which we used to generate the unwrapped meshes. The idea behind this is that once a model is unwrapped we obtain a kind of 2D map of it, and we can paint a texture on it using the unwrapped mesh (the "map") as a template. By doing this we were able to generate and assign one single texture per object. The process consisted in exporting the objects again to .obj, import them in Blender, apply a tool called *smart projections*, export them one more time to .obj and import them back to 3DS Max. Here we used *render to texture* to project the old textured meshes (the ones with several textures applied on them) on the unwrapped ones and generate a single texture. Basically, if we apply now the single image generated with render to texture to the unwrapped mesh, it will look exactly the same as the not unwrapped one.

The only thing left to do now is to use Raptor to generate the .eoz files for the objects. Sometimes the textures will be automatically generated and assigned to the mesh, and some others we will have to create a texture node in EON, link it to the material of the object and load the texture. Once the objects have been imported we can scale and rotate them if needed, and change their world coordinates to place them wherever it is convenient.



# Chapter 5

## Interacting with the virtual environment

This chapter describes in detail the process of adding interaction to our -so far lifeless- virtual museum. We will start with a brief overview of the tool used to do so, followed by the actual implementation of the navigation and the interaction with the exhibits. Next we will proceed to talk about the Nintendo Wii controller and how we can use it with a laptop or computer, and finally we will show the script used in our case to control the museum with it.

### 5.1 EON Studio

EON Reality is a 3D visual content management and Virtual Reality software provider. With their software tools and applications users are able to create realistic and authentic product experiences based on 3D visualization technology. EON Studio is one of these products and the latest version (6.1) has been used in this thesis to implement the navigation and the interaction with the museum. As the company itself describes it, "*EON Studio is a complete GUI based authoring tool for developing real-time 3D multimedia applications focused on Sales/Marketing, Training/Support and Visualization.*". Even if we do not pretend to write a manual about using it, we feel that some basics are needed to understand the current chapter. Their web site [57] can be checked for further details.

When one opens EON Studio they find a number of child windows within the main window, as shown in Figure 5.1. These windows are used for the different tasks they perform when constructing EON applications. An EON application is constructed by arranging and connecting basic components called nodes, which represent various features that may be included in a simulation such as a camera, a 3D object, a node for mouse-interaction, or a node for adding multimedia effects.

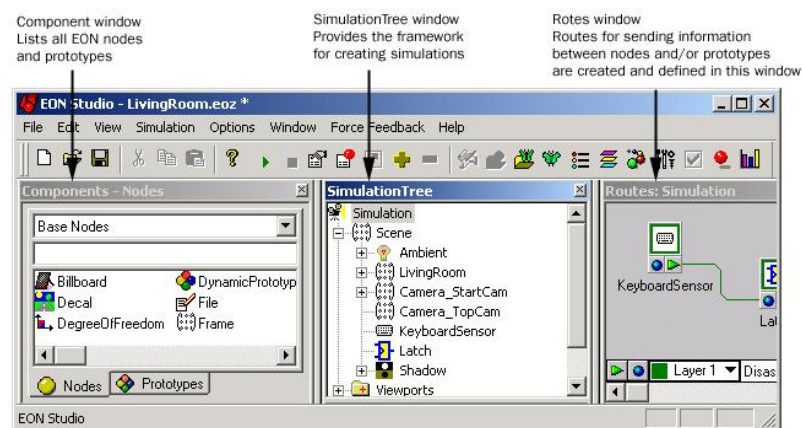


Figure 5.1: The three basic windows needed to build an EON application

The figure below describes how to work in EON Studio. These steps are performed when building all EON applications, but the work procedure in more complex systems may also include writing scripts.

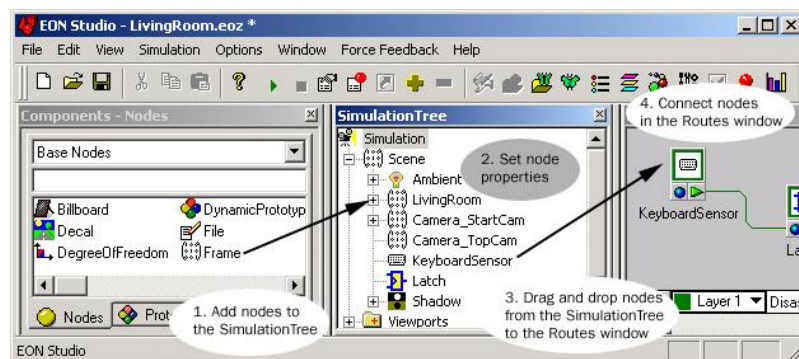


Figure 5.2: Working flow in EON Studio

## 5.2 Navigation

EON Studio implements a node specially designed to provide the walk motion models found in many 3D environments (node *Walk*). It can be controlled either with a mouse or a joystick, and by using it we can navigate and also look around the environment. In our case, though, we chose to use the *WalkAbout* node, which also lets you navigate within the simulation using the keyboard instead. We thought that with this node we would have more freedom and it would make it easier to map the Wiimote input.

Once we were able to walk and look around, we wanted to add more realism to the scene and include collision detection. There are several ways to do this in EON Studio, but basically we tried two of them: the node *Avatar Collision* and the collision provided by the *WalkAbout* node itself. Since *Avatar Collision* was slowing down the simulation a lot, we chose to activate the collision in the *WalkAbout* node, even if we noticed some problems with it. For example, the pictures on the walls became kind of portals which randomly teleport the user to another point of the museum. Nevertheless, in this case the simulation was still working smoothly so this was the final solution.

## 5.3 Interaction with virtual objects

### 5.3.1 Selection

The first task we should be able to perform is picking up an object, the one we would like to interact with. Our first try was to use the *ISector* node, which emits a sensor ray along the Y-axis and detects when the ray is intersected by *Mesh* node objects. If placed beneath a *Camera* node, the *ISector* can be used to detect specified objects that are in front of it. The problem with this was that the camera had to be placed right in front of the object so it was detected, and we thought it would be better to detect the objects when being approached from any direction. Moreover it became a bit complicated to identify which of the objects was the intersected one.

After taking a second look at the list of components we found the *ProximitySensor*. This node is activated when the camera is positioned within a specified radius, and it can send events that may be used by other nodes. Since this was exactly what we wanted to do we decided to place a *Proxim-*

*itySensor* node under the node corresponding to every mesh. This method solved at the same time two problems: 1. Detecting an object without being in front of it, and 2. Identify the object which is detected, since each exhibit has its own sensor, and every sensor emits its own events.

Now the goal was to give the visitor some feedback about it. We needed to show in some way that there is an object within the proximity range and that we can pick it up if we want. A good solution for this was to show a label every time we get close to an object, indicating its name and the action to be performed in case we want to select it. At this point we noticed that we could not do something like that by using only routes between nodes, and felt the need to introduce our first script, called *InspectionMode*.

The first step in the creation of the script was to create some fields and connections between the *Script* node and the proximity sensors. We created an *inevent* field in the Script for each object (called *select[Name of the object]*), and connected the *InProximityTrue outevent* field of every *ProximitySensor* node to the corresponding field in the script. Also a field called *selected*, with type *field*, was added to the script with the purpose of storing an identifier for the object within the range. Basically, when the camera approaches an object, the *ProximitySensor* placed in it sends an *InProximityTrue outevent*, which is caught by the corresponding *select[Name of the object] inevent* in the script. The function associated to each of these *inevents* updates the value of the field *selected* with the number used to identify the object. Figure 5.3 in the next page shows the aforementioned routes.

In order to show a label indicating the name of the object we used a *2DText node*. This node is updated every time a *ProximitySensor* throws an *InProximityTrue* event, and hidden when the event thrown is *InProximityFalse*.

### 5.3.2 Inspection mode

When we talk about *inspection mode* we are referring to the status of the simulation in which the user can interact with an object. This means rotating the exhibit to inspect it (if the exhibit is not a picture), reading some related information and playing a sound in some cases. The activation of this mode is triggered by a *KeyboardSensor* node associated to the key 'Q'. The *KeyboardSensor* is only enabled when we are within the proximity range of any of the exhibits, otherwise pressing down the key 'Q' produces no effect.

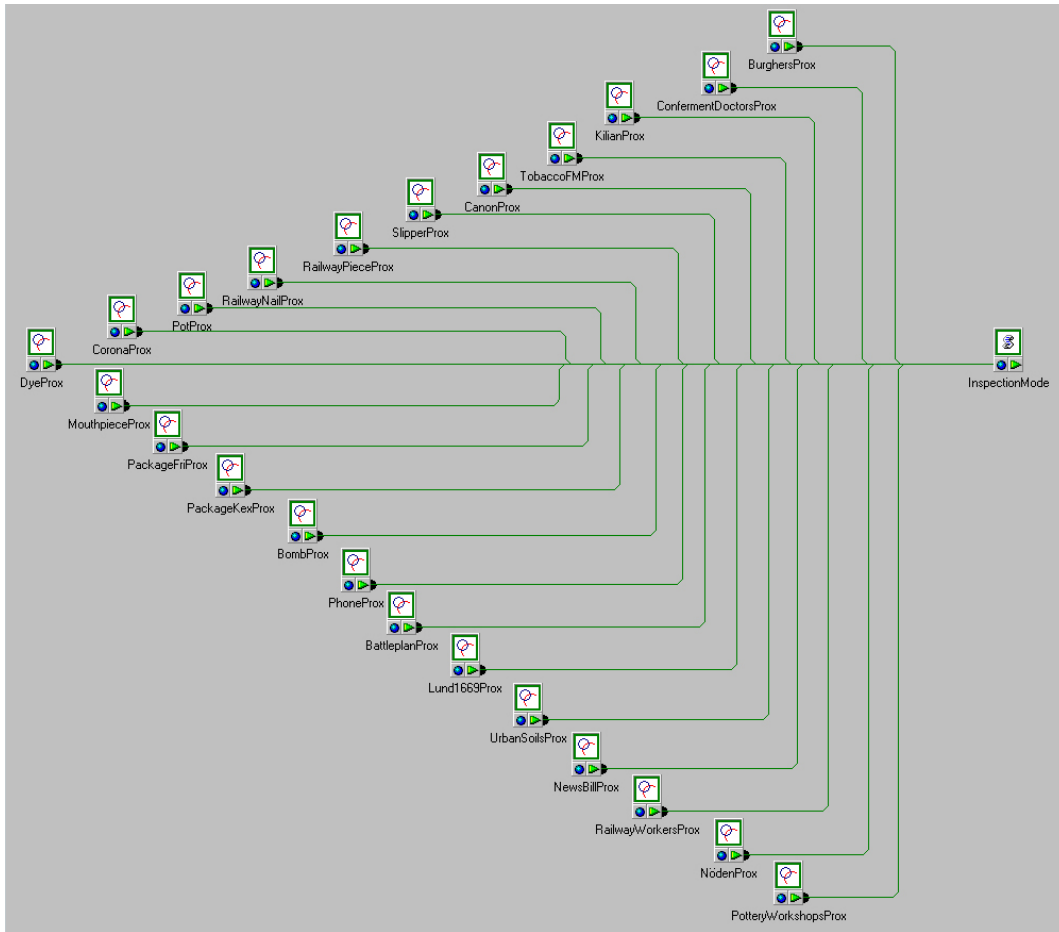


Figure 5.3: Routes used to connect the *ProximitySensor* nodes to the script

When we switch to this mode we want to focus the visitor's attention into the selected exhibit. We achieve this by directing the camera towards it and zooming in, and by hiding everything but the walls of the museum and the object itself. The camera is placed in its new location in a smooth way by using a *Place* node. The process is quite simple: we have a list with predefined coordinates and orientation for each object. Depending on the identifier stored in the field *selected* we update the camera position to the one which is predefined for that object. The *Place* node calculates a trajectory starting at the current position of the camera and ending at the predefined one. Also, the *WalkAbout* node is disabled so we cannot move the camera.

To be able to rotate the objects we use six *Rotate* nodes per object (one per axis and direction), which we enable when entering the inspection mode.



These nodes are connected to six *KeyboardSensors* so that the press of one of these keys produces rotation of the object in the corresponding axis and direction.

When switching to inspection mode there are also some changes in the GUI. Most of the objects have associated texts or sounds that should be shown or played while inspecting the object. A label placed on the bottom left corner of the screen allows the user to visualize the text, and another label on the bottom right corner can be used to play the sound. These labels are *2DText* nodes connected to two *KeyboardSensors* (associated to the keys 'T' and 'G' respectively) in such a way that when the user presses the key, either the text is shown or the sound is played. By pressing 'T'/'G' again the text is hidden or the sound is stopped. These actions are controlled by a second script called *ExhibitInfo*, which has the information about which objects have associated texts and which ones have sounds. Alternatively these labels can be pressed by using the mouse cursor and the left button. The components used for the texts and sounds are *2DImage* and *DirectSound* nodes. The *KeyboardSensors* have effect only when we are in inspection mode, and only if there is a text and/or a sound available for the selected object.

Furthermore, a label on the top left corner of the screen indicates the user that it is possible to exit inspection mode by pressing B. Again, we use a *2DText* node connected to a *KeyboardSensor*. When we switch back to navigation mode we would like to restore everything and leave it just as it was before inspection mode was entered. What we have to do is to reset the object to its original position, unhide all the objects and labels we hid before, stop the sound if it was playing, unenable and disable the corresponding *KeyboardSensors*, disable the *Rotation* nodes, restore the camera position and orientation and enable the *WalkAbout* node.

Finally, a *KeyboardSensor* connected to the 'Home' key can be used at any time to restore the camera. If we press it while in navigation mode the camera will be restored to the simulation's initial position. If it is pressed while inspecting an object, the object's orientation will be restored to the initial one.

The complete code can be found in Section 11.2

### 5.3.3 The Wii controllers

Changing topics, it is time to talk about one of the novelties of this thesis which is the use of the Nintendo Wii controllers as input device. We will not discuss here why this particular input device was chosen, since it has been already discussed in previous work on interaction techniques in virtual museums[4].

As stated before, the Wii Remote (also known as Wiimote) and Nunchuck are the controllers for the Nintendo Wii console (see Figure 5.4). A main feature of the Wii Remote is its motion sensing capability, which allows the user to interact with and manipulate items by moving the device and pointing on the screen. This is possible thanks to the inclusion of accelerometers and optical sensors in its design. If we want to use the Wii Remote as an accurate pointing device we need to make use of a sensor bar, which is used to locate the cursor when the device is moved up, down, left and right. It also allows to calculate the distance between the Wii Remote and the sensor bar, so it can also control forward-backward motion of an object in a 3D environment. Other features include a basic audio system and rumble, and also the possibility to attach expansions.

One of this expansions is the Nunchuck controller, which connects to the Wii Remote via a cord. Like the Wii Remote, the Nunchuk provides a three-axis accelerometer for motion-sensing and tilting, some extra buttons and a joystick[60].



Figure 5.4: Nintendo Wii Remote (on the right) and Nunchuck (on the left) controllers

## BlueSoleil

BlueSoleil (see Figure 5.5) is one of the most popular Windows-based Bluetooth stack, available for Windows XP, Windows Vista, Windows Mobile and Linux among others. It makes it possible to wirelessly access a wide variety of Bluetooth enabled digital devices from a computer, such as mobile phones, headsets, printers or keyboards. In our case we needed to access the Wii controllers and this has been the application used for that. For further information one can refer to the official website[58].

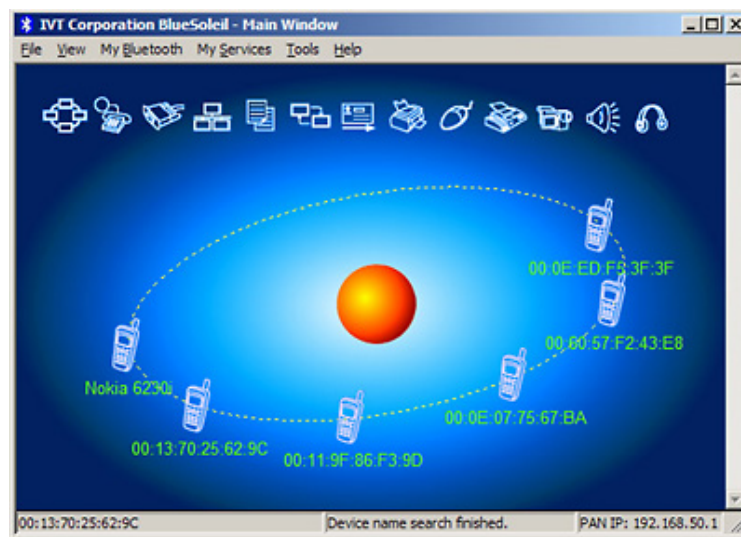


Figure 5.5: BlueSoleil main window

## GlovePIE

GlovePIE (see Figure 5.6) stands for Glove Programmable Input Emulator. It was originally started as a system for emulation of joystick and keyboard input using the Essential Reality P5 Glove, but now it supports emulating all kinds of input (keyboard, mouse, multiple joysticks and gamepads, MIDI devices, etc), using all kinds of devices (multiple mice, keyboards, joysticks, virtual reality gloves, a microphone, etc). One of the devices it offers support to are the Wii controllers[59].

Because it is free, easy-to-use and it has a good reputation, it has been the tool selected to map the Wii controllers input.

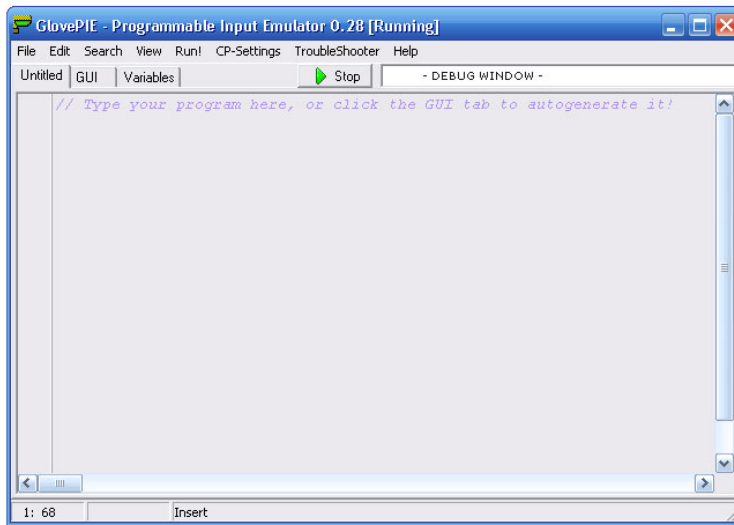


Figure 5.6: GlovePIE main window

## Mapping the input

The emulation of the keyboard and the mouse in GlovePIE is as easy as assigning the corresponding button in the input device to the desired key/button. For example, if we want the button A of the Wiimote to have the same effect as the key A in the keyboard when pressed, we should write in our script something like this:

```
Keyboard.A = Wiimote.A
```

We could emulate the mouse in the same way, for example:

```
Mouse.LeftButton = Wiimote.A
```

To access the Nunchuck and map the joystick to the arrow keys on the keyboard we could type something like the following:

```
Keyboard.Up = Wiimote.Nunchuk.JoyY < -0,5  
Keyboard.Down = Wiimote.Nunchuk.JoyY > 0,5  
Keyboard.Left = Wiimote.Nunchuk.JoyX < -0,5  
Keyboard.Right = Wiimote.Nunchuk.JoyX > 0,5
```

We can now complicate it a bit more and make the cursor move according to where the Wiimote is pointing:

```
Mouse.DirectInputX = 1023-(wiimote.dot1x+wiimote.dot2x)/2
Mouse.DirectInputY = (Wiimote.dot1y+Wiimote.dot2y)/2
```

Next you can find the list of correspondences between Wiimote/Nunchuck and keyboard/mouse, and also the action performed in the museum. IM indicates that the action has effect only in inspection mode, WM indicates walk mode:

- Q (keyboard) → A (Wiimote) → Select exhibit(WM)
- T (keyboard) → 1 (Wiimote) → Show information (IM)
- G (keyboard) → 2 (Wiimote) → Play sound (IM)
- Arrows (keyboard) → Joystick (Nunchuck) → walk and look around (WM)
- Home (keyboard) → Home (Wiimote) → Reset camera (WM) / Reset object position (IM)
- U, J, I, K (keyboard) → Roll and pitch (Wiimote) → Rotations of the objects (+X, -X, +Y, -Y axis) (IM)
- O, L (keyboard) → Left and right (Wiimote) → Rotations of the objects (+Z, -Z axis) (IM)
- Volume up (keyboard) → + (Wiimote) → Turns up the volume
- Volume down (keyboard) → - (Wiimote) → Turns down the volume
- Page up (keyboard) → Z + Joystick (-Y axis) (Nunchuck) → Look up (WM)
- Page down (keyboard) → Z + Joystick (+Y axis) (Nunchuck) → Look down (WM)
- Left button (mouse) → A (Wiimote) → Show information or play sound when clicking on the corresponding label in the GUI (IM)
- Mouse motion → Infra-red sensor (Wiimote) → Cursor

The GlovePIE code used for the mapping can be found in Section 11.3

# Chapter 6

## Usability test

This virtual museum has been designed keeping in mind some user requirements, as described in Section 3. We would like to make sure that these requirements have been fulfilled, so the application can actually be used by the public in a real museum. The analysis described next has the aim to test the usability of the museum and measure our successfulness in some way.

### 6.1 Standards in usability and user-centered design

There is a large number of published guidelines and standards related to this matter, which can be divided into three main categories[53]:

1. Product usage characteristics: how well users perform with it (Satisfaction)
2. Product interface attributes: design of the interface and interaction
3. Development process: activities carried out during product development

Within all the standards and guidelines included in the first two groups, which are the ones directly related to the end-user, we will use as baseline those specifications set out by ISO (the International Standards Organization), which is the main body of standards in this and many other areas.

We will focus on ISO 9241-11, which defines usability as *the extent to which a product can be used by specified users to achieve specified goals with effectiveness (fit for purpose), efficiency (work or time required to perform a task) and satisfaction (how pleasant it is to use) in a specified context of use.* (However we consider it will be enough to keep in mind this definition instead of following all the specifications in a strict way.)

## 6.2 Objectives

In our specific context *effectiveness*, *efficiency* and *satisfaction* can be applied to the following aspects:

1. Input device
  - Are the Wiimote and the Nunchuck easy to use?
  - Are they comfortable?
2. Interaction
  - Is the mapping of the controls intuitive?
  - Is the GUI user-friendly?
3. Presentation of the content
  - Is the visual aspect of the museum pleasant to the eye? Is it realistic?
  - Is the quality of the exhibits good enough? Are they realistic?
  - Is the content presented in a clear way? Is the information easy to find and read?
4. Immersion
  - Does the virtual environment give the user a sense of presence?
5. Motivation
  - Does the user have fun when interacting with the application?
  - Is the content of the museum interesting?
  - Can the usage of the application be considered as a learning experience?

## 6.3 Test group

As described in Section 3.2 we count with three differentiated target groups. We would like to get some independent information about the user response in each of these groups, so we can find out whether the application is suitable to be used by any of its potential users or not. The sample used will therefore be composed by several people pertaining to the specified age ranges. Due to time restrictions, though, we will only test subjects pertaining to the "youngest" and the "oldest" groups.

Regarding the matter of what the ideal number of test subjects should be there is still a lot of discussion. In the early 1990s, Jakob Nielsen, at that time a researcher at Sun Microsystems, popularized the concept of using numerous small usability tests instead of using a large sample. Literally, *"Elaborate usability tests are a waste of resources. The best results come from testing no more than 5 users and running as many small tests as you can afford"* [54]. However there have been many claims in response of this, tested with both empirical evidence and more advanced mathematical models than the one described by Nielsen. Most researchers and practitioners today agree that, although testing five users is better than not testing at all, a sample size larger than five is required to detect a satisfying amount of usability problems.

Considering these facts and the resource limitations of this project, which is not a final product so far, the test groups have been kept quite small (three people in each target group). This is not even the amount suggested by Nielsen, but enough to get some guidance feedback regarding design flaws.

## 6.4 Testing method

The method used to carry out the tests is *user testing*, which is one the most basic and useful methods. It has three components[55]:

- Get some representative users
- Ask the users to perform representative tasks with the design
- Observe what the users do, where they succeed, and where they have difficulties with the user interface



We combine video recording of the user, subsequent interviews, and also a questionnaire. The recorded data is very useful to get corporal language feedback and spontaneous responses, and the questionnaire provides information regarding more specific details with closed-answer questions.

### 6.4.1 Hardware settings

Two cameras have been used to register the tests. One of them points directly to the test subject in a way that we can see both face and hands. The other camera registers everything that happens on the screen. Also, a microphone has been used to register the voice.

### 6.4.2 Tasks

The test consists of the following tasks:

**Task 1: Warming-up navigation** The user walks around in the museum to get familiar with the interface and the input device. We can test here how easy-to-learn the application and the input device are.

**Task 2: Finding a specific exhibit and getting related information** The user is informed to look for a certain object or picture. Once the exhibit is found we ask for some related information. This gives us some feedback related to the clarity of the content and the design of the GUI.

**Task 3: Inspect and manipulate one of the exhibits** The user gets familiar with the rotation system and the manipulation of items by moving the Wiimote.

**Task 4: Free navigation and interaction with the virtual environment** The user is asked to do whatever he/she wants to do. This can help us find unexpected behaviors, and also get some ideas about what a normal user would like or try to do.

## 6.5 Results

### 6.5.1 Questionnaire

The charts below show the results obtained from the questionnaire:

#### 1. Input device

(a) Would you consider the controllers easy to use?

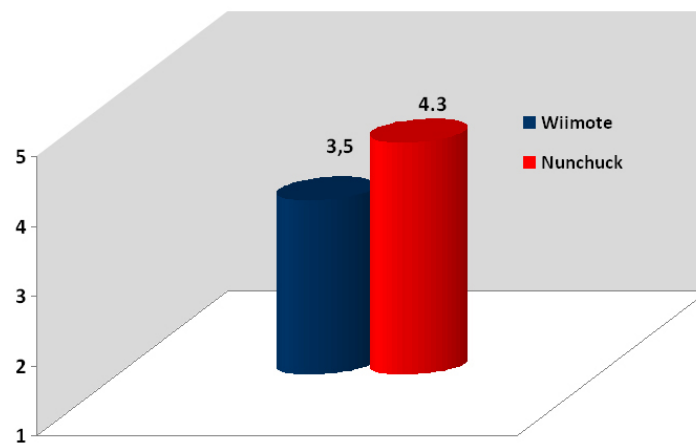


Figure 6.1: Results for usability of the input devices

(b) Did you feel comfortable when using them?

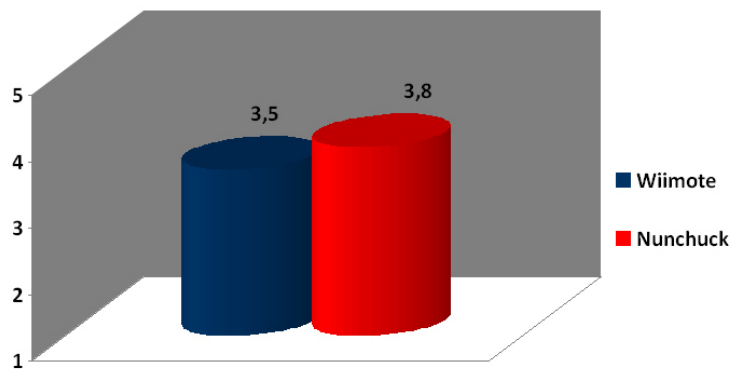


Figure 6.2: Results for comfortability of the input devices

## 2. Interaction

- (a) Do you think the action performed by each button is intuitive?
- (b) Was it easy to learn how to use the controllers?
- (c) Are the on-screen options easy to use?

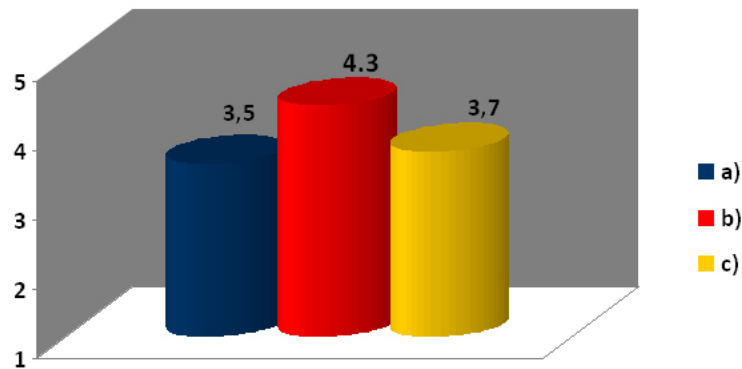


Figure 6.3: Results for interaction

## 3. Presentation of the content

- (a) Was the visual appearance of the museum nice?
- (b) Would you consider the visual appearance of the museum realistic?

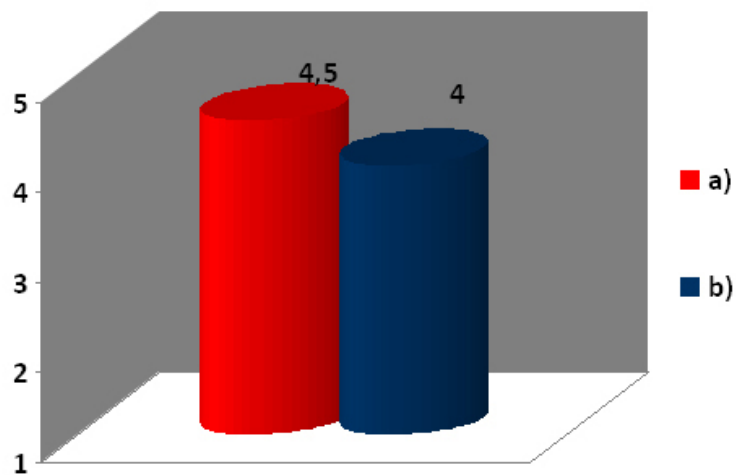


Figure 6.4: Results for presentation of the content: museum

- (c) Is the number of exhibits adequate?
- (d) Is the quality of the pictures and objects good enough?
  - i. Pictures
  - ii. Objects

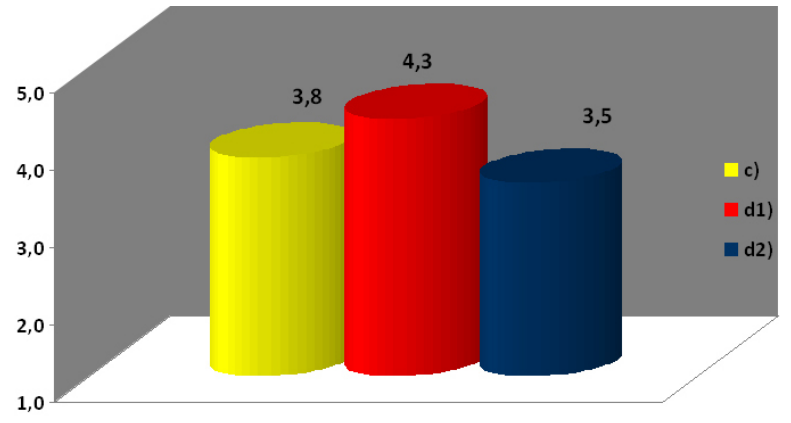


Figure 6.5: Results for presentation of the content: exhibits

- (e) Is the content presented in a clear way?
  - i. The information is easy to find
  - ii. The information is easy to read

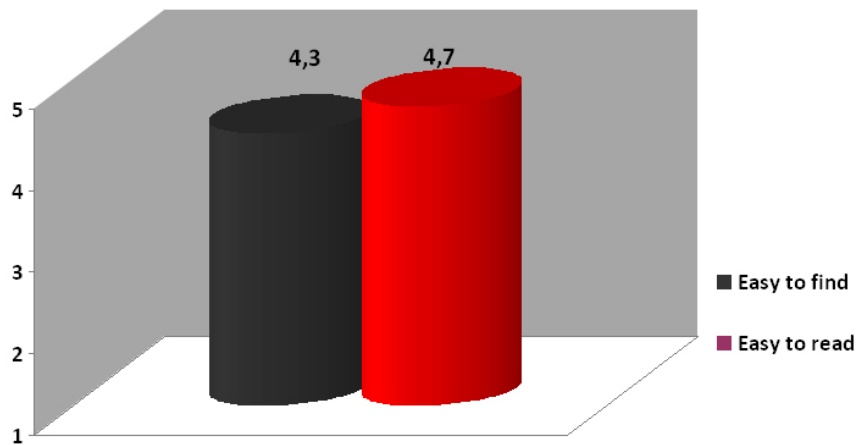


Figure 6.6: Results for presentation of the content: information

#### 4. Immersion

- (a) Did you at some moment have the feeling of being inside a real museum?

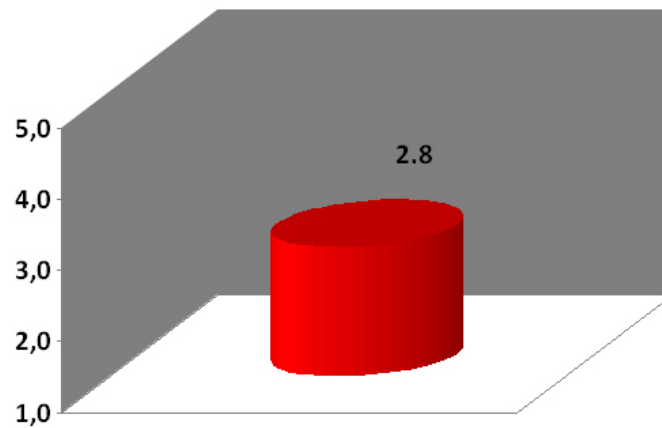


Figure 6.7: Results for immersion

#### 5. Motivation

- (a) Did you have fun while walking around the museum?  
(b) Is the content of the exhibition interesting?  
(c) Did you learn something while using the application?

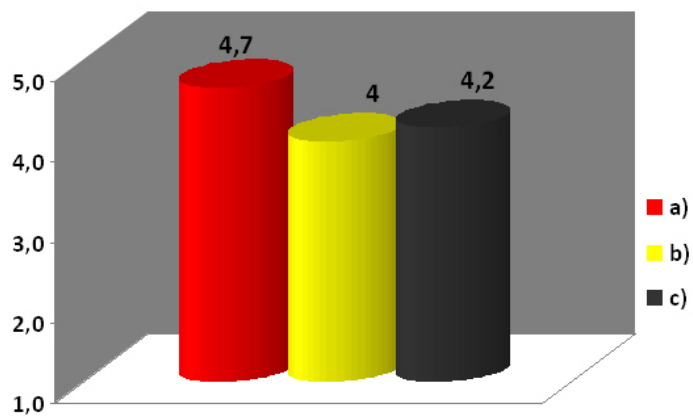


Figure 6.8: Results for motivation

## 6.5.2 Interview

We will now give an overview of the answers given during the interviews.

1. Describe briefly your experience in the museum

The subjects agreed that exploring the museum was a fun experience and that it was pleasant to explore. In most cases they felt curious to investigate the images and objects. Some of them also appreciated the lighting and the decoration. Nevertheless we find some contradictions when they give their opinion about the Wii controllers. Some of them had a good feeling when using them from the beginning, whilst others needed some time to become familiar with them and asked for a better control. Learning about the history of Lund was also a positive point for some subjects. One of the subjects thought the movement was slow.

2. Did you have any expectations before trying the application? If so, did it cover them?

The subjects did not have any special expectations, except just walking around and watching some exhibits. Hence they appreciated to have the possibility of rotating the objects as this is not possible in a real museum.

3. Which kind of related content would you like to find in a virtual museum?

Several subjects stated that more information in either textual or (preferably) multimedia way, such as more sounds or videos, would greatly improve the teaching capabilities and interest of the museum. We found also an interesting comment concerning the interaction possibilities: "I would like more interaction with virtual humans, animals, talking artifacts or game-like interactions and puzzles and stories, I realize these might come in a later implementation". On the other hand some others were completely satisfied with it, since "to smell or touch something is not really possible in a virtual program". This is actually possible with the use of haptics but we noticed that people do not really know about this.

4. Do you think there is something missing or something that can be significantly improved?

A few subjects complained about the control of the cursor since it is not easy to direct. In the "older" group we find also a suggestion about the

size of the cursor, which could be larger. Better graphics and collision detection are some other suggestions, and maybe the use of the Wiimote to look around, even if they themselves agree that it could be too shaky. The number of the exhibits could be also increased depending on the context (a stand-alone virtual museum or as a complement to a real one). We found here another very interesting comment. One of the subjects stated that the interface messages should be integrated in the environment. "This relates to the immersive quality and simplifying the psychological movement from being inside the museum and then suddenly outside reading the interface messages". One of the subjects suggested the inclusion of a guide, a voice giving information.

### 6.5.3 Observations

In this section we will enumerate some facts that have been observed during the tests. We will discuss them afterwards in Section 6.6.

#### 1. Input device

- Even if the users less used to gamepads and joysticks had some difficulties at the beginning, in general they all seemed to feel comfortable when using the controllers. The pointer of the Wiimote, though, was harder to use and all of them preferred to press buttons instead of trying to place the cursor at some point of the screen.
- Some subjects felt like they had to hold the Wiimote in the air instead of relaxing their arms. Also, they were prone to get closer to the screen when losing track of the cursor, and very often pressed with the cursor outside the simulation screen. This seemed to make them confused since the simulation was not working anymore.

#### 2. Interaction

- Sometimes the users got stuck while walking.
- "It is not so intuitive for selecting", "It would be nice if there were some kind of boundaries so I can just do what I am doing now, change the mode". The selection method is not clear enough.

- "It is not so intuitive for rotating", "I cannot really get how that works". The users tended to move the Wiimote to the sides instead of rotating it or lifting it up, an action which causes no effect.
- Some walls can be crossed, and when the camera collides with a picture the user is teleported to another point of the museum.
- It is hard to select some exhibits when they are close to another exhibit.
- Many users find it more intuitive to select the objects with the cursor and tend to try doing this.

### 3. Presentation of the content

- The test subjects were very successful when finding the object requested in the second task, which makes us think that the indications were clear enough and the exhibits well located. Nevertheless, in some cases they did not notice the indications shown in the GUI. They were expecting to see all the information integrated in the environment as if it was part of the museum.
- "Can I sit? It is inviting", "It is a very happy ambient", "This is a nice room", "I feel very comfortable in here". These are some of the comments made by the users when walking around in the museum.
- "It doesn't say 2 here". It is true that the GUI does not tell the user that it is possible to show a text or play a sound by pressing the Wiimote's buttons 1 and 2.

### 4. Immersion

- Some of the comments made by the test subjects were "I want to smell it", "I want to talk with the girl", "Can I sit?", "Can I look at the plant?". This kind of comments show the willing of the user to have a more complete experience. They would like to have more interaction options which would give them a greater sense of realism, hence a higher degree of immersion.

### 5. Motivation

- "I think I studied the wrong stuff", "I'm a Jedi knight", "It's really cool", "Very nice", "It's awesome" (when rotating objects), "Super" (when playing a sound), "Oh, it looks great!" (when zooming



in and out of an object). The test subjects had fun while exploring the museum, specially when they were able to interact with the objects. Our guessing is that this may be due to the novelty of the input device. The sounds were also very welcome and most of the subjects laughed when listening to them.

- "That was interesting", "I had no idea" (when reading some related information). Even if most of the subjects seemed to be more interested in the visual appearance and the interaction possibilities, some of them were really interested in the information given. We can state that these subjects learned something while exploring the museum.

## 6.6 Discussion

We will now discuss the previous results and draw some conclusions. To continue with the schema followed so far we will divide these conclusions in scopes:

### 1. Input device

- We agree with the test subjects that the pointer device is not easy to control (See Graphs 6.1 and 6.2). Maybe it could be possible to make some adjustments or try with different distances and positions of the sensor bar.
- Regarding the suggestion of looking around by using the Wiimote, we think that it totally goes in contraposition with the previous statement. Since it is hard to direct we think it would only make things worse. At this moment the only feature affected by this are the texts and sounds showed in inspection mode, which can be showed and played by using the cursor. Since they can also be activated by pressing buttons it is not such a big deal, but if the navigation was also controlled like that it would make the whole museum really hard to use. It could also be tiring to have to hold the Wiimote pointing to the sensor bar all the time.

### 2. Interaction

- The fact that the application stucks sometimes could be due either to loading times or to the collision detection system. In the first

case we could try to reduce the number of polygons or the quality of the textures used. If the problem is caused by the collision then there is not much to do about it, as we are using a component provided by EON Studio. Maybe we could try to use other solutions to implement it, such as the *CollisionObject* and the *CollisionGeometryType* nodes, or the *CollisionBoxes* prototype.

- Since the collision detection does not work perfectly, some of the walls can be crossed. This is not something we can fix by now, unless we try the alternative solutions mentioned in the previous point.
- Concerning the rotations, the motion could be improved by using the Wii Motion Plus, an expansion device that allows the Wii Remote to more accurately capture complex motion. The addition of a multi-axis gyroscope is what makes this possible [56].
- We could easily avoid pointing outside the simulation window by running the simulation in full-screen. The reason we did not do this before was to prevent having the wrong appearance of the interface. As all the labels shown in the GUI are placed in a position relative to the top left corner of the screen, the resizing of the simulation window would cause the labels to appear in the wrong position. To avoid this we decided to fix its size, and hence the fullscreen mode is not available. Again, since we are using predefined EON Studio nodes, there is nothing we can do here. The perfect solution would be to have the possibility to place these labels in a position relative to the other corners of the screen.
- During the tests we noticed that the selection was not as intuitive as expected (See Graph 6.3). Probably it would be a good idea to show a box or some other indicator around or on top of the selected object, instead of using a label to indicate which is the object available for inspection.
- About the suggestion of selecting the exhibits with the pointer, even if it makes sense, we feel like the selection would become much harder. It also makes sense to select by proximity, and it forces the visitor to walk around to get close to the exhibit, so we think the current solution is preferable.
- We agree again with the subjects that some exhibits are really hard to select. The proximity sensors should be adjusted to avoid this. Another possibility is to place the objects further from each other, so the proximity sensors do not overlap.

### 3. Presentation of the content

- Keeping in mind the nice comments that the subjects made about the visual appearance of the museum, we feel in general satisfied with the result. We think we were successful in the task of building an environment that is pleasant to the eye. The quality of the exhibits was also good enough, and the environment quite realistic (see Graphs 6.4 and 6.5).
- Since nobody had problems when performing the second task (find a specified object) we also think that we managed to present the content in a clear way (See Graph 6.6).
- There is some information missing in the GUI. For example, like one of the subjects observed, there is no information telling the user that the buttons 1 and 2 can be used to play sounds and show information. This information should be added.

### 4. Immersion

- The degree of immersion observed was low (see Graph 6.7), but it was just what we expected it to be. Of course it is not easy to achieve immersion while being seated on a chair in front of a computer screen. We have some ideas here that will be discussed further in Section 8, like running the simulation in a CAVE or using haptics.
- The use of talking avatars or the integration of the GUI with the environment could also help achieving more immersion. The idea of including an audio guide is also interesting.

### 5. Motivation

- During the tests we could see the subjects laughing and imaging themselves as Jedi knights. Of course we could still significantly improve the degree of fun they had, but we feel successful in this point. We can also notice by observing the charts (see Graph 6.8) that the users enjoyed their visit in the museum, and that was one of the goals we wanted to achieve.
- Watching the museum from a different point of view, we could also notice some interest in learning about the history of the exhibits. The subjects were curious about them and could learn some new things (see Graph 6.8), hence we can consider this museum as a learning tool.

# Chapter 7

## Conclusions

The aim of this thesis was to build a virtual museum which was realistic, highly usable and interactive. As a complement, we wanted to be innovative and use an uncommon controller as input device. In our way to achieve these goals we have been more or less successful, and have also learned many things that we will try to capture in this chapter.

### 7.1 The museum

Taking a look at our work we can say we got a positive result in overall. We managed to build a museum, navigate across it, fill it with some interesting content that we can interact with and, moreover, do that by using the Wii Remote and Nunchuck. Since the basics are there we consider our expectations to be fulfilled.

Going further than the basics, though, there are some aspects which would be interesting to improve. Let's recover the requirements we established previously in this document. First, we wanted to build an application with an easy-to-use graphical interface and an easy-to-learn interaction device. Even if our interface is quite simple, and the users were able to learn how to use it quite quickly, we found out that it could be much more intuitive. We are also conscious that some information about how to interact with it is missing in the current version. Regarding the input device we are satisfied with it, even if some improvements could be done in the control of the cursor.

The next requirement was focused on the visual appearance of the museum and the interaction possibilities. Due to time constraints and some technical issues we could not generate really good meshes for all the exhibits, but we could do so with some of them. We have the feeling that we proved what is possible to do, and improving the rest of the meshes is only a matter of time. On the other hand we managed to have really good quality pictures, and the users felt comfortable with the distribution and the appearance of the museum itself. We feel therefore satisfied with the result in this aspect.

When we started to think about interaction possibilities in a virtual museum we had many ideas. Besides the most basic ones we could think about, we considered other features such as the inclusion of sound, video, games or quizzes, and even the use of a balance board as device to walk around the museum. Even so we knew that the time constraints of a thesis are not that flexible, so we established some minimum features we would like to have in our museum: to be able to walk around, inspect the exhibits and get information about them. All these tasks were available in the final version of the museum. In the end we are glad that we not only managed to fulfill the basics, but we were also able to add multimedia information as an "extra".

Another goal we wanted to fulfill is realism, both in the visual appearance and the navigation. The museum, referring to the building, could perfectly be a reproduction of a real one, hence we think we succeed in this. The navigation, though, could be improved. We designed a first person camera instead of a free camera to make the user feel like if he was actually walking in the museum, but this is not enough. For example, we did not add any restriction in the degree of freedom of the camera besides a fixed height. Therefore, the visitor is able to rotate it 180° and walk upside down, which is not realistic at all. Also, since the collision detection is not working properly we feel a bit disappointed in this aspect.

A good response time was another desirable property of the application. Although when comparing the final version with previous ones we can notice a huge improvement in the speed, during the usability tests we could see that it still stucks sometimes. As discussed previously we do not really know if this problem is due to a slow loading time or to an improper collision detection, but the point is that it sometimes keeps the motion from being smooth.

## 7.2 Technical concerns

When we started this project we were novice in the use of almost all the tools we needed. It was the first time we had to model, to use a 3D scanner, to build an application in EON Studio and to write scripts for a Wiimote. Moreover we had to use the latest versions of all the software, which is often risky as the newest versions are usually unstable and filled of bugs. This makes it also harder to find online manuals or expert people who can help understanding how the tools work. Maybe in some cases this is not a big deal, but in our case it was the cause of many problems and a huge investment of time. This is why we would like to say a word about it.

EON Studio is a very interesting tool for developing 3D interactive environments. It is very fast and easy to build a basic simulation with some interesting interaction possibilities. Anyway, the problem comes when the application starts to become more and more complex. Our experience showed us that sometimes the only solution to fix an error is to start all over again. We had to rebuild the museum twice because of some improper behaviors of the application. Also there are some small details that really slow down the development. For example, it is not possible to select several nodes at the same time in the simulation tree. This means that if we need to move several nodes from one Frame to another we have to do it one by one. When having a small simulation there is no problem, but when it has more than a hundred nodes this insignificant detail becomes a serious problem. Another annoying fact is that it is not possible to copy the routes from one .eoz file to another without having to arrange all the components in the route window again by hand. There were also some problems when trying to copy scripts, since they stopped working even if the code and the nodes used were exactly the same.

EON Raptor is another tool we had some problems with. Most of the times either the textures were not exported properly from .3ds to .eoz, or the meshes had holes. Some other times 3DS Max was crashing directly when trying to run Raptor.

The Next Engine Scan Studio did not present such big problems, but it is not easy at all to align the meshes. We earnestly recommend the use of other tools such as Blender or 3DS Max as a complement to refine them.

On the other hand we felt very comfortable when using GlovePIE and 3DS Max.



# Chapter 8

## Future Work

Previously in this document we identified several aspects of the application that could be extended or improved. We will list them now and give some guidance for future projects.

### 8.1 Selection

One of the most important points to improve, in our opinion, is the selection of objects. First of all the range of the proximity sensors should be adjusted so that they do not interfere with each other. We think a value within the range (2 , 2.5) would work. We also noticed that a label indicating the name of the closest exhibit does not give enough feedback to the user, hence we could complement it with an arrow pointing to the object or drawing a box around it.

### 8.2 Rotations

It would be interesting to check if the Wii Motion Plus produces any improvement in the way the rotations work with the current version. Furthermore, the pivot point of some of the objects should be centered.

### 8.3 Interface

Some possible improvements in the interface arose during the observation and analysis of the results. The idea of mixing the interface with the environment



would help achieving a better immersion and probably make it even more easy to use.

## 8.4 Exhibits

Another basic aspect that could be improved is the quality of the meshes. As we stated before it was not possible to create high quality meshes in some cases, but with more time they could be refined and also the textures could be better. Some more exhibits could also be added to the collection.

## 8.5 Interaction

This is one of the aspects where it is better to extend it with more possibilities instead of improving the ones that already exist. The addition of more interaction possibilities such as games, quizzes, talking avatars or videos would extend the teaching capabilities and the interest of the museum.

## 8.6 Immersion

A very interesting idea here would be to use some kind of haptic device, such as a data glove, so the visitor can actually *feel* the exhibits with his hands. Feeling force feedback is always welcome in any kind of application, but we believe that in a museum it would be even more interesting. One of the best advantages of virtual museums is the possibility to interact with the objects, since in real museums the exhibits are usually protected behind a glass. This would greatly increase the degree of immersion and interest of our application.

Another good idea would be to run the application in a CAVE. We are conscious that it is not so easy to get the hardware needed for that, or to make it fit in a real museum. Anyway we are sure that it would make a totally new and interesting experience.

# Chapter 9

## Acknowledgements

To conclude this report I would like to say a word of thanks to all the people who helped me somehow bringing this project to life:

**Konrad Tollmar** For first accepting me as master student at Lund University. Thanks to him I got in touch with the RE-FLEX flexible reality center and with my actual supervisor.

**Mattias Wallergård** My supervisor, and coordinator of RE-FLEX, who was always there to bring up new ideas and suggestions, introduce me to many interesting and helpful people, deal with all the practical stuff and help me with anything I needed. His interest and efforts encouraged and motivated me during all the project.

**Joakim Eriksson** The go-to-guy for me at the Virtual Reality lab. His diligence and experience with 3DStudio Max and EON Studio were very helpful and appreciated, and also his efforts to help me with many other issues.

**Stefan Lindgren** - My contact at the Humanities Lab. He took care of giving me access to the laser scanner and helped me in every way he could.

**Carolina Larsson** - For her help with modeling. She built the model of the canon and showed me some tricks to improve my own models, which was extremely helpful in one of the most critical moments of the project.

**John Wentworth** For all the funny conversations and crazy ideas. To have such an experienced and motivated person around is always welcome, and in my case it was not only helpful for my thesis but also interesting at a personal level.

**Gertie Eriksson and Sebastian Goksor** - Our contacts at Kulturen. They gently provided all the exhibits shown in the museum and showed much interest in this project. Gertie was responsible for writing the information related to them, acquiring the pictures and take care of all the practical issues.

**Everyone who participated in the usability test** I cannot forget to thank all the people who volunteered to participate in the tests. This part of the project was very important and their feedback was really useful, specially in regards to future work.

**My friends in Lund** - Specially Markus Eger, Sabine Weber, Alina Belova and Marco Leitgeb. For their support, caring, patience and help during these months.

**My family and friends back home** - For their comprehension, love and support, and for giving me the chance to have the time of my life. I will never forget the many great experiences I had while being here, all the fantastic people I met and the great amount of things I learnt. Coming to Lund was the best decision I could have taken, and this would not have been possible without my family.

# Chapter 10

## References

- [1] Science Encyclopedia, <http://science.jrank.org/pages/7196/Virtual-Reality-origin-virtual-reality.html>, as visited in november 2008
- [2] The Board of Trustees of the University of Illinois, <http://archive.ncsa.uiuc.edu/Cyberia/VETopLevels/VR.History.html>, as visited in november 2008
- [3] Tschritzis, D., and Gibbs, S., 1991. *Virtual museums and virtual realities*. In: Proceedings of the International Conference on Hypermedia and Interactivity in Museums, Pittsburgh, PA, October 1991
- [4] *Comparing interaction techniques in a virtual reality museum framework*, master thesis by Dirk Verhagen, 2008
- [5] Maria Lorena Lehman, <http://sensingarchitecture.com/2009/02/13/applying-virtual-reality-to-museum-exhibits>, as visited in march 2009
- [6] Hemminger, B., Bolas, G., Carr, D., Jones, P., Schiff, D., and England, N., 2004. *Capturing Content for Virtual Museums: from Pieces to Exhibits*
- [7] Dr. Bradley Hemminger, School of Information and Library Science, <http://ils.unc.edu/bmh/virseum>, as visited in march 2009
- [8] Johan Olsson, Marie Mattson, and Rolf Stenbert, <http://www.reflex.lth.se/culture/kulturen/drotten>, as visited in march 2009
- [9] Kulturen Museum, <http://www.kulturen.com>, as visited in march 2009

- [10] Johan Olsson, Marie Mattson, and Rolf Stenbert, [http://www.reflex.lth.se/culture/files/Drotten/Dokument/Drotten\\_X-rappKulturen.pdf](http://www.reflex.lth.se/culture/files/Drotten/Dokument/Drotten_X-rappKulturen.pdf), as visited in march 2009
- [11] Chun-Rong, H., Chu-Song, Ch., and Pau-Choo, Ch., 2005. *Tangible photorealistic virtual museum*
- [12] Asano, T., and Ishibashi, Y. , 2004. *Adaptive Display Control of Exhibits in a Distributed Haptic Museum*
- [13] Lepouras, G., Katifori, A. , Vassilakis, C., and Charitos, D., 2003. *Real exhibitions in a virtual museum*
- [14] Wang, L., Yang, C., Qi, M., Wang, R., Meng, X., and Wang, X., 2006. *Design of a Walkthrough System for Virtual Museum Based on Voronoi Diagram*
- [15] Wei, L., Dinghao, Z., and Jingui, P., 2008. *An XML-Based Scene Description Language for 3D Virtual Museum* . In: Proceedings of the ITI 2008 30th Int. Conf. on Information Technology Interfaces, June 23-26, 2008, Cavtat, Croatia
- [16] Lepouras, G., and Vassilakis, C., 2003. *Virtual museums for all - employing game technology for edutainment*
- [17] Johnny Chung Lee, <http://johnnylee.net/projects/wii>, as visited in march 2009
- [18] Johnny Chung Lee, <http://www.youtube.com/user/jcl5m>, as visited in march 2009
- [19] Johnny Chung Lee, <http://www.wiimoteproject.com>, as visited in march 2009
- [20] Hasse Muller, <http://wii.hasse.nl>, as visited in march 2009
- [21] Re-Flex flexible reality centre, <http://www.reflex.lth.se>, as visited in march 2009
- [22] Kay Dekker, [http://en.wikipedia.org/w/index.php?title=Virtual\\_reality&direction=prev&oldid=250858617](http://en.wikipedia.org/w/index.php?title=Virtual_reality&direction=prev&oldid=250858617), as visited in november 2008
- [23] Slater, M. and Wilbur, S., 1997. *A framework for immersive virtual environments (FIVE): Speculations on the role of presence in virtual environments. Presence Teleoperators and Virtual Environments*, Vol. 6, no. 6, pages 603-616

- [24] Merriam Webster Online Dictionary, <http://www.merriam-webster.com/dictionary/haptic>, as visited in november 2008
- [25] Wordreference, <http://www.wordreference.com/definition/simulation>, as visited in november 2008
- [26] <http://www.designnews.com/media/52-PC2140125537.mp3>, as visited in november 2008
- [27] Daniel Fllman Department of Informatics Ume University, <http://www.informatik.umu.se/~dfallman/projects/vrie>, as visited in november 2008
- [28] John Schaffer, <http://www.newhorizons.org/strategies/technology/shaffer.htm>, as visited in november 2008
- [29] Sunrise, <http://www.sunrisevr.com/vrlearn.htm>, as visited in november 2008
- [30] Jönsson, P., Wallergård, M., Karlson, B., Johansson, G., Österberg, K., Eek, F., and Eriksson, J. *Trier Social Stress Test in the Black Box: Inducing Social Stress in a Virtual Environment*, Poster presented at A Brain Research Meeting: Stress, Coping and Disease, Arlington, Washington, USA, 13 - 14 November 2008.
- [31] Bram Stolk, Faizal Abdoelrahman, Anton Koning, Paul Wielinga, Jean-Marc Neefs, Andrew Stubbs, An de Bondt, Peter Leemans, and Peter van der Spek, <http://portal.acm.org/citation.cfm?id=569673.569676>, as visited in november 2008
- [32] Juan C. Drsteler , [www.infovis.net/printMag.php?num=81&lang=2](http://www.infovis.net/printMag.php?num=81&lang=2), as visited in november 2008
- [33] Wgungfu, [http://en.wikipedia.org/w/index.php?title=Battlezone\\_\(1980\\_video\\_game\)&oldid=253345987](http://en.wikipedia.org/w/index.php?title=Battlezone_(1980_video_game)&oldid=253345987), as visited in november 2008
- [34] W Industries/Virtuality, <http://www.amigahistory.co.uk/virtuality.html>, as visited in november 2008
- [35] <http://secondlife.com>
- [36] David K. Elson, and Mark O. Riedl, <http://www.aaai.org/Papers/AIIDE/2007/AIIDE07-002.pdf>, as visited in november 2008

- [37] NationMaster, <http://www.nationmaster.com/encyclopedia/Virtual-cinematography>, as visited in november 2008
- [38] [http://en.wikipedia.org/w/index.php?title=Virtual\\_cinematography&oldid=249560449](http://en.wikipedia.org/w/index.php?title=Virtual_cinematography&oldid=249560449), as visited in november 2008
- [39] Bruce Newman Mercury News, [http://www.temple.edu/ispr/examples/ex03\\_05\\_13b.html](http://www.temple.edu/ispr/examples/ex03_05_13b.html), as visited in november 2008
- [40] Dragice, [http://en.wikipedia.org/w/index.php?title=3D\\_Interaction&oldid=236500739](http://en.wikipedia.org/w/index.php?title=3D_Interaction&oldid=236500739), as visited in november 2008
- [41] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev, *3D user interfaces. Theory and practice*, Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, 2004
- [42] SharkD, [http://en.wikipedia.org/w/index.php?title=Interaction\\_technique&oldid=266696014](http://en.wikipedia.org/w/index.php?title=Interaction_technique&oldid=266696014), as visited in january 2009
- [43] Benjamin Dominic, [http://en.wikipedia.org/w/index.php?title=Six\\_degrees\\_of\\_freedom&oldid=266162749](http://en.wikipedia.org/w/index.php?title=Six_degrees_of_freedom&oldid=266162749), as visited in january 2009
- [44] Pisharov <http://en.wikipedia.org/w/index.php?title=Multi-touch&oldid=265825858>, as visited in january 2009
- [45] Mikeblas, <http://en.wikipedia.org/w/index.php?title=Touchscreen&oldid=266546281>, as visited in january 2009
- [46] Lightbot, <http://en.wikipedia.org/w/index.php?title=Powerwall&oldid=253331607>, as visited in january 2009
- [47] Dancter, [http://en.wikipedia.org/w/index.php?title=Head-mounted\\_display&oldid=265371545](http://en.wikipedia.org/w/index.php?title=Head-mounted_display&oldid=265371545), as visited in january 2009
- [48] Robbot, [http://en.wikipedia.org/w/index.php?title=Cave\\_Automatic\\_Virtual\\_Environment&oldid=257801349](http://en.wikipedia.org/w/index.php?title=Cave_Automatic_Virtual_Environment&oldid=257801349), as visited in december 2008
- [49] Immersion Corporation, [http://www.immersion.com/corporate/press-room/what\\_is\\_haptics.php](http://www.immersion.com/corporate/press-room/what_is_haptics.php), as visited in december 2008
- [50] TechTarget, [http://searchcio-midmarket.techtarget.com/sDefinition/0,,sid183\\_gci212226,00.html](http://searchcio-midmarket.techtarget.com/sDefinition/0,,sid183_gci212226,00.html), as visited in december 2008
- [51] Novint Technologies, [http://home.novint.com/products/novint\\_falcon.php](http://home.novint.com/products/novint_falcon.php), as visited in january 2009

- [52] SensAble Technologies, <http://www.sensable.com/haptic-phantom-desktop.htm>, as visited in january 2009
- [53] Usability Partners Stockholm, <http://www.usabilitypartners.se/usability/standards.shtml>, as visited in march 2009
- [54] Jakob Nielsen, <http://www.useit.com/alertbox/20000319.html>, as visited in march 2009
- [55] Jakob Nielsen, <http://www.useit.com/alertbox/20030825.html>, as visited in march 2009
- [56] Frank Caron, <http://arstechnica.com/gaming/news/2008/08/wii-motion-sensor.ars/2>, as visited in march 2009
- [57] EON Reality, <http://www.eonreality.com>, as visited in february 2009
- [58] BlueSoleil, <http://www.bluesoleil.com>, as visited in february 2009
- [59] Carl Kenner, <http://carl.kenner.googlepages.com/glovepie>, as visited in february 2009
- [60] Dancter, [http://en.wikipedia.org/w/index.php?title=Wii\\_Remote&oldid=273533484](http://en.wikipedia.org/w/index.php?title=Wii_Remote&oldid=273533484), as visited in february 2009





# Chapter 11

## Appendices

### 11.1 List of exhibits

The complete collection of exhibits shown in the museum is listed below, classified by scopes:

- Lund becomes Swedish, the city during the 17th and 18th centuries
  - Picture: Battle plan by Erik Dahlberg
  - Picture: Map of Lund from the year 1669
  - Picture: The urban soils
- The University
  - Object: Conferment canon
  - Picture: Photograph of Kilian Stobaeus
  - Picture: Photograph of the ceremony of conferment of doctor's degrees in Lund the 31 may 1900
- The inhabitants of Lund
  - Object: Slipper from an inhabitant in Nden
  - Picture: Photograph of Stora Fiskaregatan by Jns Mrtensson in 1909
  - Picture: Photograph of a burgher
- Small-scale industry in Lund
  - Object: Wrongly burnt pot from Mns Markusson Wallerus

- Object: Clich from the dye factory of Borg KM 18700
- Object: 3 commercial packages
- Object: Package from a tobacco company in Malm
- Object: Cigarette mouthpiece
- Object: Headphone for a local phone
- Picture: Pottery workshops in Lund
- Medical history
  - (information only)
- When the railroad came to Lund
  - Object: Railway nail
  - Object: Piece of rail
  - Picture: Photograph of railway workers
- Second World War
  - Object: The big bomb
  - Picture: News bill from 1943

## 11.2 Code of the EON Studio Scripts

### 11.2.1 InspectionMode

```
function initialize()
{
    // Called when simulation starts. Initialization code

    eon.SetScriptTimeout(0)
    eon.MessageBox("The screen resolution should be 1024x768 for
this simulation to work properly.\nPlease change your configuration
if your settings are different than this.", "Warning")
    eon.SetScriptTimeout(10)

preamble.value = false
inspection.value = false
selected.value = -1
eon.FindNode("PressA").GetfieldByName("IsActive").value = false
eon.FindNode("PressB").GetfieldByName("IsActive").value = false
eon.FindNode("A").GetfieldByName("IsActive").value = false
eon.FindNode("B").GetfieldByName("IsActive").value = false
eon.FindNode("TextButton").GetfieldByName("IsActive").value = false
eon.FindNode("SoundButton").GetfieldByName("IsActive").value = false
eon.FindNode("T").GetfieldByName("IsActive").value = false
eon.FindNode("G").GetfieldByName("IsActive").value = false
eon.FindNode("WalkAbout").GetfieldByName("isActive").value = true
eon.FindNode("PlaceCamera").GetfieldByName("Stop").value = true

//Initial camera
pos=eon.FindNode("ActiveViewportCamera").GetfieldByName("Position").value.toArray()
orient=eon.FindNode("ActiveViewportCamera").GetfieldByName("Orientation").value.toArray()
camera.SetMFCount(0)
camera.AddMFElement(eon.MakeSFVec3f(pos[0], pos[1], pos[2]))
camera.AddMFElement(eon.MakeSFVec3f(orient[0], orient[1], orient[2]))
eon.FindNode("PlaceCamera").GetfieldByName("TypeH").value = 1
eon.FindNode("PlaceCamera").GetfieldByName("TypeP").value = 1
eon.FindNode("PlaceCamera").GetfieldByName("TypeR").value = 1
eon.FindNode("PlaceCamera").GetfieldByName("TypeX").value = 1
eon.FindNode("PlaceCamera").GetfieldByName("TypeY").value = 1
eon.FindNode("PlaceCamera").GetfieldByName("TypeZ").value = 1

//List of objects
nrObj.value = 13
nrPic.value = 10
objects.SetMFCount(0)
objects.AddMFElement("RailwayNail")
objects.AddMFElement("RailwayPiece")
objects.AddMFElement("PackageFri")
objects.AddMFElement("PackageKex")
```

```

objects.AddMFElement("TobaccoFM")
objects.AddMFElement("Mouthpiece")
objects.AddMFElement("Pot")
objects.AddMFElement("Corona")
objects.AddMFElement("Bomb")
objects.AddMFElement("Slipper")
objects.AddMFElement("Dye")
objects.AddMFElement("Phone")
objects.AddMFElement("Canon")
objects.AddMFElement("NewsBill")
objects.AddMFElement("RailwayWorkers")
objects.AddMFElement("Kilian")
objects.AddMFElement("ConfermentDoctors")
objects.AddMFElement("PotteryWorkshops")
objects.AddMFElement("Burghers")
objects.AddMFElement("Nden")
objects.AddMFElement("UrbanSoils")
objects.AddMFElement("Lund1669")
objects.AddMFElement("BattlePlan")

//Init object nodes and rotations
for(i=0; i<nrObj.value; i++)
{
eon.FindNode(objects.GetMFElement(i)+"Global").GetfieldByName("Hidden").value = false
eon.FindNode(objects.GetMFElement(i)+"Local").GetfieldByName("Hidden").value = true
eon.FindNode(objects.GetMFElement(i)+"RotXP").GetfieldByName("active").value = false
eon.FindNode(objects.GetMFElement(i)+"RotXN").GetfieldByName("active").value = false
eon.FindNode(objects.GetMFElement(i)+"RotYP").GetfieldByName("active").value = false
eon.FindNode(objects.GetMFElement(i)+"RotYN").GetfieldByName("active").value = false
eon.FindNode(objects.GetMFElement(i)+"RotZP").GetfieldByName("active").value = false
eon.FindNode(objects.GetMFElement(i)+"RotZN").GetfieldByName("active").value = false
}

eon.FindNode("JXN").GetfieldByName("isActive").value = false
eon.FindNode("KYN").GetfieldByName("isActive").value = false
eon.FindNode("LZN").GetfieldByName("isActive").value = false
eon.FindNode("UXP").GetfieldByName("isActive").value = false
eon.FindNode("IYP").GetfieldByName("isActive").value = false
eon.FindNode("OZP").GetfieldByName("isActive").value = false

//Predefined cameras for object inspection (pos, orient)
cameras.SetMFCount(0)
texts.SetMFCount(0)
sounds.SetMFCount(0)
//RailwayNail
cameras.AddMFElement(eon.MakeSFVec3f(4.8, -6.3, 0.75))
cameras.AddMFElement(eon.MakeSFVec3f(90, 0, 0))
//RailwayPiece
cameras.AddMFElement(eon.MakeSFVec3f(5, -7.1, 0.6))

```

```

cameras.AddMFElement(eon.MakeSFVec3f(90, 0, 0))
//PackageFri
cameras.AddMFElement(eon.MakeSFVec3f(11.6, -1.1, 0.65))
cameras.AddMFElement(eon.MakeSFVec3f(180, 0, 0))
//PackageKex
cameras.AddMFElement(eon.MakeSFVec3f(10.2, -1.1, 0.7))
cameras.AddMFElement(eon.MakeSFVec3f(180, 0, 0))
//TobaccoFM
cameras.AddMFElement(eon.MakeSFVec3f(5.8, -1, 0.75))
cameras.AddMFElement(eon.MakeSFVec3f(180, 0, 0))
//Mouthpiece
cameras.AddMFElement(eon.MakeSFVec3f(6.68, -1.3, 0.67))
cameras.AddMFElement(eon.MakeSFVec3f(180, 0, 0))
//Pot
cameras.AddMFElement(eon.MakeSFVec3f(14.1, -1.3, 0.65))
cameras.AddMFElement(eon.MakeSFVec3f(180, 0, 0))
//Corona
cameras.AddMFElement(eon.MakeSFVec3f(10.9, -1.2, 0.73))
cameras.AddMFElement(eon.MakeSFVec3f(180, 0, 0))
//Bomb
cameras.AddMFElement(eon.MakeSFVec3f(4.3, -15.5, 0.6))
cameras.AddMFElement(eon.MakeSFVec3f(180, 0, 0))
//Slipper
cameras.AddMFElement(eon.MakeSFVec3f(1.5, 11.02, 0.5))
cameras.AddMFElement(eon.MakeSFVec3f(90, 0, 0))
//Dye
cameras.AddMFElement(eon.MakeSFVec3f(12.3, 1, 0.7))
cameras.AddMFElement(eon.MakeSFVec3f(0, 0, 0))
//Phone
cameras.AddMFElement(eon.MakeSFVec3f(6.9, 1, 0.7))
cameras.AddMFElement(eon.MakeSFVec3f(0, 0, 0))
//Canon
cameras.AddMFElement(eon.MakeSFVec3f(-1, -13.05, 0.65))
cameras.AddMFElement(eon.MakeSFVec3f(270, 0, 0))
//NewsBill
cameras.AddMFElement(eon.MakeSFVec3f(3, -15.8, 1.5))
cameras.AddMFElement(eon.MakeSFVec3f(90, 0, 0))
//RailwayWorkers
cameras.AddMFElement(eon.MakeSFVec3f(4.2, -6.1, 1.45))
cameras.AddMFElement(eon.MakeSFVec3f(180, 0, 0))
//Kilian
cameras.AddMFElement(eon.MakeSFVec3f(1.1, -15.6, 1.45))
cameras.AddMFElement(eon.MakeSFVec3f(270, 0, 0))
//ConfermentDoctors
cameras.AddMFElement(eon.MakeSFVec3f(0.8, -9.75, 1.5))
cameras.AddMFElement(eon.MakeSFVec3f(270, 0, 0))
//PotteryWorkshops
cameras.AddMFElement(eon.MakeSFVec3f(12.7, 1.5, 1.4))
cameras.AddMFElement(eon.MakeSFVec3f(90, 0, 0))

```

```

//Burghers
cameras.AddMFElement(eon.MakeSFVec3f(-1.4, 13.8, 1.4)) -0.6 -0.8
cameras.AddMFElement(eon.MakeSFVec3f(90, 0, 0))
//Nden
cameras.AddMFElement(eon.MakeSFVec3f(-1.2, 8.5, 1.4))
cameras.AddMFElement(eon.MakeSFVec3f(90, 0, 0))
//UrbanSoils
cameras.AddMFElement(eon.MakeSFVec3f(1.9, 15.88, 1.4))
cameras.AddMFElement(eon.MakeSFVec3f(270, 0, 0))
//Lund1669
cameras.AddMFElement(eon.MakeSFVec3f(0.9, 7.1, 1.42))
cameras.AddMFElement(eon.MakeSFVec3f(270, 0, 0))
//BattlePlan
cameras.AddMFElement(eon.MakeSFVec3f(1.8, 10.9, 1.4))
cameras.AddMFElement(eon.MakeSFVec3f(270, 0, 0))
}

function inspectionPreamble()
{
    // Show options to start Inspection Mode
    if(inspection.value == false)
    {
        if (selected.value > -1)
        {
            eon.FindNode("PressA").GetfieldByName("Text").value="Press A to select "+
                objects.GetMFElement(selected.value)
            eon.FindNode("PressA").GetfieldByName("isActive").value = true
            eon.FindNode("A").GetfieldByName("isActive").value = true
            preamble.value = true
        }
    }
}

function On_hidePreamble()
{
    // Hide options which allow to start Inspection Mode
    eon.FindNode("PressA").GetfieldByName("IsActive").value = false
    eon.FindNode("A").GetfieldByName("isActive").value = false
    preamble.value = false
}

function On_enableInspection()
{
    if (preamble.value == true)
    {
        //Disable ProxSensors to avoid "selected" to be updated
        for(i=0; i<nrObj.value+nrPic.value; i++)
        {
            obj = objects.GetMFElement(i)

```

```

        eon.FindNode(obj+"Prox").GetfieldByName("SetRun").value = false
    }

    eon.FindNode("WalkAbout").GetfieldByName("isActive").value = false
    eon.FindNode("PressA").GetfieldByName("IsActive").value = false
    enableInsp2 = true
    setCamera() //Set predefined camera
}
}

function On_enableInspection2()
{
    if (enableInsp2 == true)
    {
        enableInsp2 = false
        preamble.value = false
        //Enable and show buttons
        eon.FindNode("B").GetfieldByName("isActive").value = true
        eon.FindNode("PressB").GetfieldByName("IsActive").value = true

        //Enable rotations
        obj = objects.GetMFEElement(selected.value)
        eon.FindNode("JXN").GetfieldByName("isActive").value = true
        eon.FindNode("KYN").GetfieldByName("isActive").value = true
        eon.FindNode("LZN").GetfieldByName("isActive").value = true
        eon.FindNode("UXP").GetfieldByName("isActive").value = true
        eon.FindNode("IYP").GetfieldByName("isActive").value = true
        eon.FindNode("OZP").GetfieldByName("isActive").value = true

        ltexts = eon.FindNode("ExhibitInfo").GetfieldByName("texts")
        if (ltexts.GetMFEElement(selected) != "none")
        {
            eon.FindNode("TextButton").GetfieldByName("isActive").value = true
            eon.FindNode("T").GetfieldByName("isActive").value = true
        }

        lsounds = eon.FindNode("ExhibitInfo").GetfieldByName("sounds")

        if (lsounds.GetMFEElement(selected) != "none")
        {
            eon.FindNode("SoundButton").GetfieldByName("isActive").value = true
            eon.FindNode("G").GetfieldByName("isActive").value = true
        }

        //Hide/Unhide objects
        hideAll()
        obj = objects.GetMFEElement(selected.value)

        if(selected.value < nrObj.value)//Is an object

```



```

    {
        eon.FindNode(obj+"Local").GetfieldByName("Hidden").value = false
    }
    else //Is a picture
    {
        eon.FindNode(obj).GetfieldByName("Hidden").value = false
    }

    inspection.value=true
}
}

function On_disableInspection()
{
    if(inspection.value == true)
    {
        //Hide buttons
        eon.FindNode("TextButton").GetfieldByName("isActive").value = false
        eon.FindNode("SoundButton").GetfieldByName("isActive").value = false
        eon.FindNode("TextButton").GetfieldByName("Text").value = "SHOW INFO"
        eon.FindNode("SoundButton").GetfieldByName("Text").value = "PLAY SOUND"
        eon.FindNode("T").GetfieldByName("isActive").value = false
        eon.FindNode("G").GetfieldByName("isActive").value = false
        eon.FindNode("PressB").GetfieldByName("isActive").value = false
        eon.FindNode("B").GetfieldByName("isActive").value = false
        //Disable rotations
        eon.FindNode("JXN").GetfieldByName("isActive").value = false
        eon.FindNode("KYN").GetfieldByName("isActive").value = false
        eon.FindNode("LZN").GetfieldByName("isActive").value = false
        eon.FindNode("UXP").GetfieldByName("isActive").value = false
        eon.FindNode("IYP").GetfieldByName("isActive").value = false
        eon.FindNode("OZP").GetfieldByName("isActive").value = false

        //Hide/Unhide objects
        if(selected.value < nrObj.value)//Is an object
        {
            obj = objects.GetMFEElement(selected)
            //Reset position
            eon.FindNode(obj+"Local").GetfieldByName("Orientation").value =
                eon.FindNode(obj+"Global").GetfieldByName("Orientation").value
        }
        showAll()
        restoreCamera()//Restore previous camera
        eon.FindNode("WalkAbout").GetfieldByName("isActive").value = true

        //Stop sound/Hide text
        sound = eon.FindNode("ExhibitInfo").GetfieldByName("sound").value
        eon.FindNode(sound).GetfieldByName("Stop").value = true
        eon.FindNode("ExhibitInfo").GetfieldByName("soundOn").value = false
    }
}

```

```

    text = eon.FindNode("ExhibitInfo").GetfieldByName("text").value
    eon.FindNode("Texts!" + text + "Text").GetfieldByName("IsActive").value = false
    eon.FindNode("ExhibitInfo").GetfieldByName("infoOn").value = false

    //Enable ProxSensors to let "selected" be updated
    for(i=0; i<nrObj.value+nrPic.value; i++)
    {
        obj = objects.GetMFElement(i)
        eon.FindNode(obj+"Prox").GetfieldByName("SetRun").value = true
    }

    inspection.value = false
}
}

function setCamera()
{
    //Save current camera
    pos = eon.FindNode("ActiveViewportCamera").GetfieldByName("Position").value.toArray()
    orient = eon.FindNode("ActiveViewportCamera").GetfieldByName("Orientation").value.toArray()
    camera.SetMFElement(0, eon.MakeSFVec3f(pos[0], pos[1], pos[2]))
    camera.SetMFElement(1, eon.MakeSFVec3f(orient[0], orient[1], orient[2]))

    //Set predefined camera
    pos = cameras.GetMFElement(selected*2).toArray()
    orient = cameras.GetMFElement((selected*2)+1).toArray()

    eon.FindNode("PlaceCamera").GetfieldByName("Translation").value =
        eon.MakeSFVec3f(pos[0], pos[1], pos[2])
    eon.FindNode("PlaceCamera").GetfieldByName("Rotation").value =
        eon.MakeSFVec3f(orient[0], orient[1], orient[2])
    eon.FindNode("PlaceCamera").GetfieldByName("Start").value = true
}

function restoreCamera()
{
    eon.FindNode("PlaceCamera").GetfieldByName("Translation").value = camera.GetMFElement(0)
    eon.FindNode("PlaceCamera").GetfieldByName("Rotation").value = camera.GetMFElement(1)
    eon.FindNode("PlaceCamera").GetfieldByName("Start").value = true
}

function hideAll()
{
    eon.FindNode("Museum!Tables").GetfieldByName("Hidden").value = true

    for(i=0; i<nrObj.value; i++)
    {
        obj=objects.GetMFElement(i)
        eon.FindNode(obj+"Global").GetfieldByName("Hidden").value = true
    }
}

```

```

    }

    //Hide pictures
    for(i=nrObj.value; i<(nrObj.value+nrPic.value);i++)
    {
        obj = objects.GetMFEElement(i)
        eon.FindNode("Pictures!" +obj).GetfieldByName("Hidden").value = true
    }
}

function showAll()
{
    //Show museum
    eon.FindNode("Museum!Tables").GetfieldByName("Hidden").value = false

    //Show all obj
    for(i=0; i<nrObj.value;i++)
    {
        obj = objects.GetMFEElement(i)
        eon.FindNode(obj+"Global").GetfieldByName("Hidden").value = false
        eon.FindNode(obj+"Local").GetfieldByName("Hidden").value = true
    }

    //Show pictures
    for(i=nrObj.value; i<(nrObj.value+nrPic.value);i++)
    {
        obj = objects.GetMFEElement(i)
        eon.FindNode("Pictures!" +obj).GetfieldByName("Hidden").value = false
    }
}

function On_reset()
{
    //Reset local object position
    if(inspection.value == true)
    {
        obj = objects.GetMFEElement(selected.value)
        eon.FindNode(obj+"Local").GetfieldByName("Orientation").value =
            eon.MakeSFVec3f(0, 0, 0)
    }
    else//Reset camera position
    {
        eon.FindNode("ActiveViewportCamera").GetfieldByName("Orientation").value =
            eon.MakeSFVec3f(90, 0, 0)
        eon.FindNode("ActiveViewportCamera").GetfieldByName("Position").value =
            eon.MakeSFVec3f(0, 0, 1.2)
    }
}
}

```

```

function On_selectRailwayNail()
{
    if (inspection.value == false)
    {
        selected.value = 0
        inspectionPreamble()
    }
}

function On_selectRailwayPiece()
{
    if (inspection.value == false)
    {
        selected.value = 1
        inspectionPreamble()
    }
}

function On_selectPackageFri()
{
    if (inspection.value == false)
    {
        selected.value = 2
        inspectionPreamble()
    }
}

function On_selectPackageKex()
{
    if (inspection.value == false)
    {
        selected.value = 3
        inspectionPreamble()
    }
}

function On_selectTobaccoFM()
{
    if (inspection.value == false)
    {
        selected.value = 4
        inspectionPreamble()
    }
}

function On_selectMouthpiece()
{
    if (inspection.value == false)
    {

```

```

        selected.value = 5
        inspectionPreamble()
    }
}

function On_selectPot()
{
    if (inspection.value == false)
    {
        selected.value = 6
        inspectionPreamble()
    }
}

function On_selectCorona()
{
    if (inspection.value == false)
    {
        selected.value = 7
        inspectionPreamble()
    }
}

function On_selectBomb()
{
    if (inspection.value == false)
    {
        selected.value = 8
        inspectionPreamble()
    }
}

function On_selectSlipper()
{
    if (inspection.value == false)
    {
        selected.value = 9
        inspectionPreamble()
    }
}

function On_selectDye()
{
    if (inspection.value == false)
    {
        selected.value = 10
        inspectionPreamble()
    }
}

```

```

}

function On_selectPhone()
{
    if (inspection.value == false)
    {
        selected.value = 11
        inspectionPreamble()
    }
}

function On_selectCanon()
{
    if (inspection.value == false)
    {
        selected.value = 12
        inspectionPreamble()
    }
}

function On_selectNewsBill()
{
    if (inspection.value == false)
    {
        selected.value = 13
        inspectionPreamble()
    }
}

function On_selectRailwayWorkers()
{
    if (inspection.value == false)
    {
        selected.value = 14
        inspectionPreamble()
    }
}

function On_selectKilian()
{
    if (inspection.value == false)
    {
        selected.value = 15
        inspectionPreamble()
    }
}

function On_selectConfermentDoctors()
{

```

```

        if (inspection.value == false)
        {
            selected.value = 16
            inspectionPreamble()
        }
    }

function On_selectPotteryWorkshops()
{
    if (inspection.value == false)
    {
        selected.value = 17
        inspectionPreamble()
    }
}

function On_selectBurghers()
{
    if (inspection.value == false)
    {
        selected.value = 18
        inspectionPreamble()
    }
}

function On_selectNden()
{
    if (inspection.value == false)
    {
        selected.value = 19
        inspectionPreamble()
    }
}

function On_selectUrbanSoils()
{
    if (inspection.value == false)
    {
        selected.value = 20
        inspectionPreamble()
    }
}

function On_selectLund1669()
{
    if (inspection.value == false)
    {
        selected.value = 21
        inspectionPreamble()
    }
}

```

```

    }
}

function On_selectBattlePlan()
{
    if (inspection.value == false)
    {
        selected.value = 22
        inspectionPreamble()
    }
}

```

## 11.2.2 ExhibitInfo

```

    function initialize()
    {
// Called when simulation starts
// Add initialization code
infoOn.value = false
soundOn.value = false
eon.FindNode("TextButton").GetfieldByName("Text").value = "SHOW INFO"
eon.FindNode("SoundButton").GetfieldByName("Text").value = "PLAY SOUND"

//List of texts
nrObj.value = 23
texts.SetMFCount(0)
texts.AddMFElement("RailwayNail")
texts.AddMFElement("RailwayPiece")
texts.AddMFElement("PackageFri")
texts.AddMFElement("PackageKex")
texts.AddMFElement("TobaccoFM")
texts.AddMFElement("Mouthpiece")
texts.AddMFElement("Pot")
texts.AddMFElement("Corona")
texts.AddMFElement("Bomb")
texts.AddMFElement("none")//Slipper
texts.AddMFElement("Dye")
texts.AddMFElement("Phone")
texts.AddMFElement("Canon")
texts.AddMFElement("NewsBill")
texts.AddMFElement("RailwayWorkers")
texts.AddMFElement("Kilian")
texts.AddMFElement("Conferment")
texts.AddMFElement("none")//PotteryWorkshops
texts.AddMFElement("Burghers")
texts.AddMFElement("Nden")
texts.AddMFElement("UrbanSoils")
texts.AddMFElement("Lund1669")

```



```

texts.AddMFElement("BattlePlan")

//List of sounds
sounds.SetMFCount(0)
sounds.AddMFElement("Train1")//RailwayNail
sounds.AddMFElement("Train2")//RailwayPiece
sounds.AddMFElement("none")//PackageFri
sounds.AddMFElement("none")//PackageKex
sounds.AddMFElement("none")//TobaccoFM
sounds.AddMFElement("none")//Mouthpiece
sounds.AddMFElement("none")//Pot
sounds.AddMFElement("none")//Corona
sounds.AddMFElement("Bomb1")//Bomb
sounds.AddMFElement("none")//Slipper
sounds.AddMFElement("none")//Dye
sounds.AddMFElement("Telephone1")//Phone
sounds.AddMFElement("none")//Canon
sounds.AddMFElement("Bomb2")//NewsBill
sounds.AddMFElement("none")//RailwayWorkers
sounds.AddMFElement("none")//Kilian
sounds.AddMFElement("none")//ConfermentDoctors
sounds.AddMFElement("none")//PotteryWorkshops
sounds.AddMFElement("none")//Burghers
sounds.AddMFElement("none")//Nden
sounds.AddMFElement("none")//UrbanSoils
sounds.AddMFElement("none")//Lund1669
sounds.AddMFElement("none")//BattlePlan

    for(i=0; i<nrObj.value; i++)
    {
        aux = texts.GetMFElement(i)
        if(aux != "none")
        {
            eon.FindNode("Texts!"+aux+"Text").GetfieldByName("IsActive").value
                = false
        }
    }
}

function On_showText()
{
    if(infoOn.value == false)
    {
        eon.FindNode("TextButton").GetfieldByName("Text").value = "HIDE INFO"
        text.value = texts.GetMFElement(eon.FindNode("InspectionMode").
            GetfieldByName("selected").value)

        if(text.value != "none")
        {

```

```

        eon.FindNode("Texts!" + text.value + "Text").GetfieldByName("IsActive").value
            = true
        infoOn.value = true
    }
}
else
{
    eon.FindNode("TextButton").GetfieldByName("Text").value = "SHOW INFO"
    eon.FindNode("Texts!" + text.value + "Text").GetfieldByName("IsActive").value
        = false
    infoOn.value = false
}
}

function On_playSound()
{
    if(soundOn.value == false)
    {
        eon.FindNode("SoundButton").GetfieldByName("Text").value = "STOP SOUND"
        sound.value = sounds.GetMFElement(eon.FindNode("InspectionMode").
            GetfieldByName("selected").value)
        eon.FindNode(sound.value).GetfieldByName("Start").value = true
        soundOn.value = true
    }
    else
    {
        eon.FindNode("SoundButton").GetfieldByName("Text").value = "PLAY SOUND"
        eon.FindNode(sound.value).GetfieldByName("Stop").value = true
        soundOn.value = false
    }
}
}

```

## 11.3 GlovePie script

```
wait 30 ms
```

```
//This is to check whether the nunchuck is detected or not
```

```
if wiimote.HasNunchuk
```

```
    wiimote.Leds = 1
```

```
else
```

```
    wiimote.Leds = 0
```

```
//WALK and LOOK AROUND
```

```
//Keyboard-Cross
```

```
Keyboard.Up = Wiimote.Up //Walk forward
```

```
Keyboard.Down = Wiimote.Down //Walk backward
```

```
Keyboard.Left = Wiimote.Left //Look left
```

```
Keyboard.Right = Wiimote.Right //Look right
```

```
//Keyboard-Nunchuk
```

```
Keyboard.Up = Wiimote.Nunchuk.JoyY < -0,5 //Walk forward
```

```
Keyboard.Down = 0,5 < Wiimote.Nunchuk.JoyY //Walk backward
```

```
Keyboard.Left = Wiimote.Nunchuk.JoyX < -0,5 //Look left
```

```
Keyboard.Right = 0,5 < Wiimote.Nunchuk.JoyX //Look right
```

```
Keyboard.PageUp = (Wiimote.Nunchuk.JoyY < -0,5) and Wiimote.Nunchuk.ZButton //Look up
```

```
Keyboard.PageDown = (0,5 < Wiimote.Nunchuk.JoyY) and Wiimote.Nunchuk.ZButton //Look down
```

```
//CURSOR
```

```
Mouse.CursorVisible = true
```

```
Mouse.DirectInputX = 1023-(wiimote.dot1x+wiimote.dot2x)/2
```

```
Mouse.DirectInputY = (Wiimote.dot1y+Wiimote.dot2y)/2
```

```
Mouse.LeftButton = Wiimote.A
```

```
//INTERACTION and OBJECT INSPECTION
```

```
Keyboard.Q = Wiimote.A //Selection
```

```
Keyboard.B = Wiimote.B //Exit
```

```
Keyboard.T = Wiimote.One //Show text
```

```
Keyboard.G = Wiimote.Two //Play sound
```

```
Keyboard.U = Wiimote.Roll > 10 degrees //Rotate X axis (Roll)
```

```
Keyboard.J = Wiimote.Roll < -10 degrees //Rotate X axis (Roll)
```

```
Keyboard.I = Wiimote.Pitch > 10 degrees //Rotate Y axis (Pitch)
```

```
Keyboard.K = Wiimote.Pitch < -10 degrees //Rotate Y axis (Pitch)
```

```
Keyboard.O = Wiimote.Left //Rotate Z axis (Yaw)
```

```
Keyboard.L = Wiimote.Right //Rotate Z axis (Yaw)
```

```
Keyboard.Home = Wiimote.Home //Reset
```

```
//VOLUME
```

```
Keyboard.VolumeUp = Wiimote.Plus
```

```
Keyboard.VolumeDown = Wiimote.Minus
```

## 11.4 Questionnaire and interview for the usability test

### 11.4.1 Questionnaire

Name:

Age:

Sex:

Occupation:

#### GLOBAL EVALUATION

Please indicate the extent to which you agree or disagree with the following statements:

- 1 - Strongly disagree
- 2 - Disagree
- 3 - Neutral
- 4 - Agree
- 5 - Strongly agree

#### 1. Input device

(a) Would you consider the controllers easy to use?

- Wiimote (right hand controller) 1 2 3 4 5
- Nunchuck (left hand controller) 1 2 3 4 5

(b) Did you feel comfortable when using them?

- Wiimote 1 2 3 4 5
- Nunchuck 1 2 3 4 5

#### 2. Interaction

(a) Do you think the action performed by each button is intuitive?  
1 2 3 4 5

(b) Was it easy to learn how to use the controllers? 1 2 3 4 5

(c) Are the on-screen options easy to use? 1 2 3 4 5

#### 3. Presentation of the content

(a) Was the visual appearance of the museum nice? 1 2 3 4 5

- (b) Is the quality of the exhibits good enough?
  - Pictures 1 2 3 4 5
  - Objects 1 2 3 4 5
- (c) Is the number of exhibits adequate? 1 2 3 4 5
- (d) Would you consider the visual appearance of the museum realistic?  
1 2 3 4 5
- (e) Is the content presented in a clear way?
  - The information is easy to find 1 2 3 4 5
  - The information is easy to read 1 2 3 4 5

#### 4. Immersion

- (a) Did you at some moment have the feeling of being inside a real museum? 1 2 3 4 5

#### 5. Motivation

- (a) Did you have fun while walking around the museum?  
1 2 3 4 5
- (b) Is the content of the exhibition interesting? 1 2 3 4 5
- (c) Did you learn something while using the application?  
1 2 3 4 5

### UTILITY OF VIRTUAL MUSEUMS

Please select one response for each question:

1. I believe that virtual museums
  - (a) are useful as a learning tool YES NO I don't know
  - (b) are useful as a way to preserve the cultural heritage  
YES NO I don't know
  - (c) can be entertaining YES NO I don't know
  - (d) can increase people's interest in museums  
YES NO I don't know
2. I would use this application if it was available in a real museum  
YES NO I don't know

### USER EXPERIENCE

Please select one response for each question:

1. How often do you use the following?
  - (a) Computers, PDA and similar  
Never   About once a month   About once a week  
2-6 times a week   Every day  
...and I use them to play videogames   YES   NO
  - (b) Video consoles  
Never   About once a month   About once a week  
2-6 times a week   Every day
  - (c) Other virtual reality systems (indicate which)  
Never   About once a month   About once a week  
2-6 times a week   Every day

#### **11.4.2 Interview**

Name:

Age:

Sex:

Occupation:

1. Describe briefly your experience in the museum
2. Did you have any expectations before trying the application? If so, did it cover them?
3. Which kind of related content would you like to find in a virtual museum?
4. Do you think there is something missing or that can be significantly improved?