University of Oslo
Department of Informatics

Don’t walk like an Egyptian

Coping with shared attention in a mobile 3D system

Marko Steffensen
Terje E.H. Torma

Cand. Scient Thesis

January 2003

Don’t walk like an Egyptian

Cand. Scient Thesis, University of Oslo, 2003

Illustration: Thoth, the Egyptian god of knowledge and magic, with a ComPAQ iPAQ PDA. Slightly retouched by the authors.

Abstract

This thesis presents our study of a system using real-time three-dimensional graphics on handheld computers focusing on the impact of the shared attention problem. The main objective is to investigate whether a mobile RT3D system can be both *usable* and *useful*. We postulate that one of the greater challenges to such a system is the problem of shared attention. The dynamic context of truly mobile IT use means that the user will have to share his attention between operating the system and relating to the world around him.

In order to investigate this issue we developed a prototype of a Mobile 3D system. We relied on literature studies, interviews with experts and other research of relevance. As a result we formulated a set of system requirements with the intention to cope with the shared attention problem. We then performed an experiment to test how the mobile 3D system developed was utilized and how our design choices affected shared attention among the test subjects.

The resulting system was received favourably by the test subjects. We observed the subjects obtaining a dynamic pattern of use where they located the destination of each task prior to movement, then checked while moving that they were on the right track. The attention demand of the system was not observed to be intrusive as the subjects appeared to be able to make efficient use of the system while moving. Although the system was received favourably by the test subjects, we suspect that our initial suppositions need to be revised. Our finds indicate that our focus on shared attention may have blinded us to other important factors. There were indications that the subjects had constructed a mental map of the geographical area prior to beginning movement and so did not use the system to aqcuire new information while they were walking. This interpretation leads to several needed revisions of our shared attention model.

In our discussion we argue that focusing our design on shared attention to such an extent is not necessarily ideal for M3D systems. However the start/stop paradigm of the subjects indicates the need for a dynamic interaction design: aimed at rapidly switching between high and low attention modes.

Preface

This thesis is a part of the Candidatus Scientiarum degree at the Department of Informatics, Faculty of Mathematics and Natural Sciences at the University of Oslo. The degree is a combination of attending advanced courses for a total of one and a half year and writing a thesis on self conducted research for a total of one year.

Acknowledgements

During this research we have had many people supporting us and helping us accomplish this thesis.

First of all we would like to thank our internal advisor, Anders Kluge, for all the work he has put into guiding us and discuss our work.

The company Octaga's expertise in 3D technology led to many interesting discussions and insight in complex 3D design. During our stay there, we wrote a paper on mobile multimedia from which we learnt a great deal, and got many ideas that helped us in defining this thesis.

The co-students at both MMCL (MultiMedia and Communications Lab) and Parken study room gave us support and a social dimension that helped making our stay a memorable one. Park1 forever!

Thanks to all the experiment subjects for their time and effort in helping us finding answers and insight into mobile 3D use.

Thanks to Tyr Steffensen and Steinar Henrik Johnsen for proof-reading and constructive criticism.

Thanks to Johan for many interesting discussions and paper/web layout assistance.

Thanks to Nina Meland for posing in illustration figures.

Marko Steffensen
Terje E.H. Torma
Oslo, January 2003

Contents

List of Acronyms

2D	Two-Dimensional
3D	Three-Dimensional
3G	3rd Generation networks
API	Application Programming Interface
AR	Augmented Reality
CPU	Central Processing Unit
GIS	Geographic Information System
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile communications
GUI	Graphical User Interface
HCI	Human-Computer Interaction
HMD	Head Mounted Display
HTML	HyperText Markup Language
HUD	Head Up Display
IR	InfraRed communication
IT	Information Technology
JVM	Java Virtual Machine
LoD	Level of Detail
LoS	Line of Sight
LTM	Long Term Memory
M3D	Mobile 3D
OS	Operating System
PDA	Personal Digital Assistant
PPC	Pocket PC
RT3D	Real-Time 3D
SMS	Short Message Service
STM	Short Term Memory
UMTS	Universal Mobile Telecommunications System
VR	Virtual Reality
VRML	Virtual Reality Modeling Language
WAP	Wireless Application Protocol
WLAN	Wireless Local Area Network

List of Figures

Chapter 1

Introduction

LACKING the technique of perspective, the paintings of the ancient Egyptians were flat and lifeless. It seems the current generation of handheld terminals is similarly devoid of techniques for representing the third dimension. Real time 3D (RT3D) has proven successful on stationary computers, but RT3D applications on mobile devices are rare - as of yet. Despite this, many of the components necessary for such a system are commercially available, including full 3D engines. It would appear that a system that exploits the benefits offered by RT3D in a mobile setting is overdue. Compounding this, even the very concept and use of mobility has, like the ancient Egyptian, lacked perspective. The handheld format offers users the ability to use a fully functional computer while on the move. But rather than using them while on the move, users have adopted “portable computing” where they only use their "mobile" device while they are standing or often sitting perfectly still. This negates many of the benefits offered by the format.

Programs found on mobile devices today are often straight ports of programs found on stationary PC’s. They are therefore not necessarily suited to “truly mobile use”, neither are they developed with mobile users in mind. We believe that RT3D on handheld terminals does have a future: that truly mobile 3D presents many exiting new opportunities and uses that are very different from what stationary systems can offer.

As we started work on our thesis, we decided that we wanted to create an experimental prototype of a truly mobile 3D system. It seemed clear to us that a mobile 3D system needed to take advantage of the ability to provide contextual information to the user. The example we decided on for our prototype was an interactive 3D map of the Oslo University’s Blindern campus. To create this system we borrowed ideas and methodology from the field of augmented reality. Our idea was to use an RT3D map to display additional information about the user’s environment, helping him to locate his destination. But this presented a challenge: Augmented Reality (AR) systems have demonstrated their usefulness and even become standard in some instances, like fighter pilot Head Up Displays. In contrast, our system depends on having the user relate to the reality around him and a model of it at the same time. Can users take advantage of such a system? We believe that by carefully designing the system to require only a low level of attention, one can achieve a usable and useful system.

1.1 Research questions

In summary, the inspiration for our thesis was our belief that:

> *Real time 3D on mobile terminals will be demonstrably useful: it will provide new capabilities or make the user more efficient.*

However, one of the main problems associated with mobile IT use is that of shared attention. This problem is compounded with mobile 3D systems because they are visually intensive, relatively complex and have an immersive quality. However, we believe that through the use of proper design techniques, the problem of shared attention can be mitigated to allow for a usable system. Based on this, we believe that:

> **A mobile 3D system can overcome the problem of shared attention if properly designed.**

In order to narrow our scope and outline our workflow we derived two separate research questions:

1. **How can a mobile 3D system be designed to compensate for shared attention?**
   We do not aim to find “the solution” to how an M3D system should be designed, but we hope that through a theoretical pre-study and an iterative design process, some insights can be gained. These insights are formulated as a set of design guidelines and system requirements.

2. **Can users utilize our system?**
   Once our system was designed, we put it through a field test to determine if research subjects could effectively utilize our system. The purpose of the field test was to determine whether the system is both usable and useful. The users must be able to use the system and take advantage of it in a realistic setting. Even though we made great effort to mitigate the shared attention problem, in the end, the system’s success hinges on the users. From these results we hope to determine whether M3D, as we have implemented it, has a potential for general use.

As is apparent from our secondary questions, our thesis has a distinct duality in terms of development and testing. This essential dualism is an important factor that affects our thesis throughout. In effect, we present a double argument: First we have to arrive at the guidelines and requirements for the system. Then we test the system itself in the hopes of validating the rationale behind the system design. The benefit of this combined method is that it hopefully will enable us to show a direct link between the technology and how it is used. The main pitfall is that this method can quickly become too inclusive; we risk "biting over more than we could chew". We needed to make some difficult choices in where we should focus our efforts and what areas we had to pass over quickly. First and foremost this is noticeable in our system design. Identifying general guidelines for mobile 3D design would require years of work and far more resources than we had at our disposal. Therefore we were forced to limit ourselves to a pre-study where we identified some basic premises to work from. At the end of our pre-study we had some firm convictions of what an M3D system should look like and what it should do in order to be useful. These ideas are presented in chapter 8 as a set of system requirements and general guidelines that account for the experiment scenario. We imagine that they can serve as reference for future M3D researchers to build upon or discard in light of the results of our experiment.

1.2 Pre-study

Before starting work on this thesis we worked four months for the company named Octaga (Octaga AS 2003)(then Applied Media Technologies, a subsidiary of Telenor Research and Development). Our task was to perform a study on the feasibility of RT3D on hand-held terminals. During this period we studied literature, performed interviews with experts, tested available M3D systems and attended seminars. The study resulted in an internal report called Mobile Multimedia. Working on this study was very useful to us: it introduced us to the field of M3D, gave us a good overview of current technologies and allowed us to network with experts on RT3D systems and mobility. In the work on this thesis, the experience we gained through working for Telenor was invaluable.

1.3 Emergence of the truly mobile terminal

There are many different definitions of what constitutes a mobile terminal, but in the context of this report we will equate mobile terminals with handheld terminals. That is, a computational device that is small enough to be held in a user's palm. Examples include both mobile phones and PDA's as well as more specialized devices such as Global Positioning Systems (GPS). In today's society, mobile terminals such as these are becoming more and more common. By the third quarter of 1999 there were more than 2.7 million mobile phone/pager subscriptions in Norway growing to approximately 3.3 million by the end of 2001 (Statistisk

Sentralbyrå 2003). These devices are becoming more and more powerful as well. This is especially the case with PDA's and hybrid PDA/phones. Thanks to recent technological advances, some of these devices are capable of performing tasks that until recently was only possible on desktop systems. The most impressive advances have come in the areas of computing power (handheld Central Processing Units (CPU's) in the range of several hundred MHz are common), graphics (screens with thousands of colors), wireless networks (high-speed wireless networks like GPRS) and even localization technologies (GPS and cell identification). In many ways, the differences between handheld and stationary terminals are becoming smaller. Perhaps in the future the differences in technical specifications between stationary and mobile terminals will disappear or nearly so (Beck et al. 2002, page 1).

In concert with the expanding capabilities of handheld terminals they are being used in new situations and contexts, different from those where stationary systems are employed. This has resulted in the increased popularity of applications that are well suited to these use-modes. Perhaps the clearest example of this is SMS. Yet there have been few attempts at creating a 3D graphics system that capitalizes on the mobile use-mode to give new functionality. This is intriguing when one considers the popularity of 3D graphics applications on stationary computers (as is pointed out in chapter 3). The recent advances give PDA's a potential for 3D graphics that seem to be just waiting to be explored: Increased processing power and screen quality allow handhelds to display complex 3D models, high-speed networks allow them to download needed 3D data quickly instead of storing it locally and localization technologies open possibilities for context-sensitive information systems (contextual systems).

1.3.1 The mobile terminal and RT3D

Unfortunately, the handheld format and mobile use pose their own problems to 3D systems. A well-known factor in almost all mobile IT-use is often referred to as shared attention, dual-tasking or other, related terms. In the context of this thesis we will mostly use the term *shared attention* for this phenomenon. An example of shared attention is when a person talks in a mobile phone at the same time as driving a car (Strayer & Johnston 2001). An illustration of how important shared attention problems are can be seen in the fact that the practice of using a cell phone while driving is illegal in many countries today. RT3D can potentially compound this problem by its very nature. RT3D is visually intensive, making for an engrossing experience for the user. On a handheld terminal this requires the user to focus on the screen in his hand rather than his environment. RT3D is also quite demanding on the user for the program to progress. In most RT3D application the user interfaces with the 3D world through an "avatar". The avatar is the representation of the user inside the 3D world. Most commonly, it is the avatar's viewpoint that is displayed on the screen, making the avatar itself invisible to the user. The user interacts with the system by moving the avatar around and manipulating various objects in the 3D world. This mode of interaction means that the system is dependent on continuous user input to progress. The combined effect of these features is often referred to as *immersion* and is the declared goal of the form of RT3D system known as *Virtual Reality*. In many ways, immersion becomes what we hope NOT to achieve with our system. Immersion signifies that the computer world is so engrossing as to blot out the impressions from the real world (Manovich 2001, page 16). Clearly, an immersive system would be at odds with our goal of compensating for shared attention. However, we found that the concept of "immersion" was too abstract to be of much use, so we will discuss the individual effects separately. An important part of our design process was modifying these features to comply with minimizing the shared attention problem.

1.4 Creating a prototype system

An important part of the incentive for this thesis was the opportunity to create a prototype M3D system. In order to investigate the viability of M3D we needed an example system on which to conduct an experiment. During our pre-study, we found no available M3D systems that fit our purpose. It would be possible to use a readily available 3D navigation system on a stationary computer, but then we would lose the opportunity to investigate mobile use patterns. Because of this, we decided to design a new M3D system specifically for our experiment.

We considered several options when making our example system. We found many possible system ideas that could fit with our guidelines (described below) and fulfill our system requirements (see chapter 8). We outlined systems that would aid a shopper in a mall, or provide structural information about buildings for architects or craftsmen. At the end of our pre-study, we had decided to create part of a mobile 3D navigation system.

We would implement part of an imagined, larger system that allowed users to interact with it through mobile terminals and relied on using RT3D for some tasks. Our implementation would focus on a 3D model of our university campus that could be used as an interactive map. A complete system would require features such as network connectivity, personalized information and interaction with other systems. To make the development manageable we decided to simulate some of these features and ignore others, in order to focus on the M3D aspects of the system. For a more detailed description of a complete system see chapter 8. The rationale behind this system is described more fully in chapter 7, but the main reasons for choosing a navigation system are outlined below.

Relevance

As explained in chapter 7, one of our primary system requirements was that the system should be helpful to the user. In other words, the system must provide meaningful data in a mobile context. Since mobility entails a dynamic context, it follows that our system needed to be *context sensitive*. We tried to identify activities that people usually perform while on the move. One such activity is navigation. When looking for a specific address in unfamiliar surroundings it makes sense to keep on the move. Navigation systems already exist for many handheld terminals. In addition, RT3D holds some promise in this area as well. Rather than relying on a 2D representation of the area as with a conventional map, a 3D system could display a more detailed and realistic model. This detail and realism should allow for easier recognition of buildings and features than traditional maps. This particular solution utilizes the advantages of both handheld terminals and real-time 3D technologies.

Resolution of location information

In a navigation system that relies on a graphic display to communicate directions to a user, the nature of the display is dependent on the information available to the system. Butz et. al. refer to the accuracy of available location data as *resolution of location* (Butz et al. 2001). Butz et. al. describes an interesting relationship between the accuracy of location information available to the system and the detail of its graphic display. Basically, if the system has very accurate information about the user's position and orientation its graphic display can be very simple. For instance, if the user finds himself at a crossroads and the system has accurate information about his position and orientation, a simple arrow displayed on the system's screen will suffice to indicate the correct direction. If, on the other hand, the system had less accurate information, say only that the user was located in the area around the crossroads, more detail would be necessary. In this case, the system would have to provide enough information for the user to determine his own location. The system could display the streetnames and building addresses to aid the user in getting a fix on his position. An arrow could then be displayed showing which road to choose. These two examples are shown in figure 1.1. The first case would correspond to part A) and the second to part D).

This relationship is highly relevant to our system as it indicates that more information is needed in our system because of the low resolution of location information. In our scenario the system will only have a very rough idea of the user's position and no orientation information at all. This necessitates a very detailed graphic way description. The level of detail provided by a near-photorealistic 3D model should in theory be enough to allow the user to determine his location.

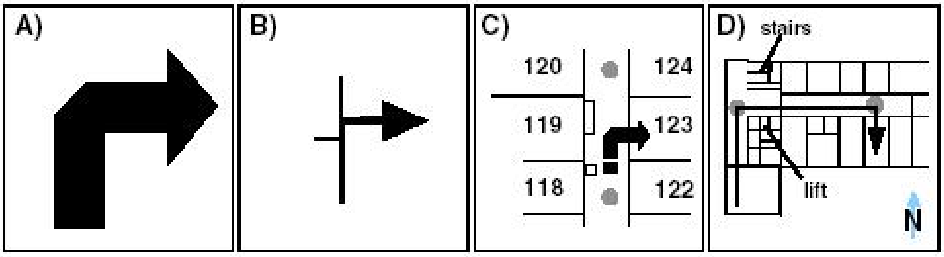

Figure 1.1: Four different graphical way description schemata that depend on the quality of orientation and position information (Butz et al. 2001). In A) a simple arrow is adequate for navigation if the resolution of location information is very high - e.g. compass and accurate location detection. As the resolution becomes less accurate, more information is necessary to display for a user in order to help him navigate.

Location detection

As described in section 4.5, accurate location detection remains a difficult technological prospect. The currently available technologies have relatively large margins of error, forcing developers of mobile navigation systems to compensate by using other technologies (as in (Butz et al. 2001)). A class of systems that is particularly affected by location resolution is known as *augmented reality systems*. The aim of such systems is to display computer supplied information overlaid on our perception of reality. Our system has a similar aim to AR in that both seek to supply additional information, linked to real world objects. In our system this information takes the form of navigation aids. In AR, the computer supplied information can be almost anything. Producing a true blend of the virtual and the real, as in AR, requires very high location and orientation resolution. On the other hand, displaying the contextual information on a separate 3D model and relegating the task of accurate location detection to the user can provide an alternative means of linking computer supplied information with real world objects. This also opens for a wide range of other possibilities as shown in the next paragraph.

Location detection as a basic function

When evaluating different system ideas, we found that many of them relied on accurate location detection. In many cases, navigation and recognition forms a basis on which to build a more specialized system. Having established his location, a user could call upon the system to display useful information about his surroundings. This information could vary between systems, but an example might include the age and history of noteworthy buildings in a tourist information system. The context-sensitive information provided need not be in the form of annotations, but could serve as reminders (i.e. "Your rent is due and there’s a bank a block from here").

1.4.1 Scenario outline

Once we had decided on creating a navigational system, we designed a scenario for its use. In our experiment, we intended to recreate this scenario with the research subjects playing the roles of users. Since we only had the resources to create a partial system, many parts of a complete system had to be simulated or simply left out. Our intention was to investigate mobile 3D use in a realistic setting, so as long as the simulated parts were transparent to the subjects, this should pose no problems. See chapters 8 and 9 for a more complete description of the implemented and simulated system components.

What follows is a description of the type of scenario we envision for a full version of our test system. We picture a Bluetooth transmitter located at a convenient landmark. Preferably this would be at a location where many people would start a search, such as a bus stop. Downloading a 3D map and additional information about the surrounding area to a PDA would be very quick and perhaps even automatic. Upon start-up, the system displays the 3D world from a viewpoint corresponding to the user's. Using the nearby landmark for reference, the user recognizes the viewpoint. The user then navigates simultaneously in the real world and the 3D model towards his destination.

The system provides valuable advice along the way, aiding the user visually with the realistic 3D model and providing helpful information interwoven with the 3D data. This information might be in the form of textual messages, such as the names and addresses of buildings, lighted paths or symbols. The system would know the initial position of the user to within the range of a short range network transmitter; the task of more accurate and dynamic positioning is relegated to the user.

In chapter 9 we describe how we modified this scenario for new students at the Blindern campus.

1.4.2 Design guidelines

We decided to perform a qualitative experiment outside a laboratory setting to test the usability of our prototype system. This was mainly because we believed a laboratory setting would be too restricted. Based on this we formulated a set of general guidelines for the system development phase. These guidelines summed up our ideas and assumptions and allowed us to maintain our focus throughout the design and testing activities.

Wide target audience

In order to say something about the potential of mobile 3D, our system must have a wide target audience. Designing a system that works for a very specific task that would only be of interest to a small group would make it hard to generalize on the potential of M3D for other groups. When we discussed possible designs we considered creating M3D systems that would aid architects or sewer workers. These concepts were discarded because we considered them too specific in their target demographics. To use the analogy of augmented reality, we can see that AR is successfully used in a few, narrow fields, but there are no general-interest AR applications readily available today. We wish to show that M3D has potential not just in a few specialized fields, but has potential for aiding in common, everyday tasks.

Off-the-shelf system components

We decided to build the system using commercially available components wherever possible. This had two benefits: First, it would support our argument that M3D is a possibility using current technology. Second, it would allow us to minimize the time and cost of development. Since we only had two people working on the project, this was of great help. We wished to show that M3D does not require any breakthroughs in technology to make it work. We had found some interesting 3D engines and viewers designed for handheld systems, and we were confident that these would allow us to quickly create a viable test system. We also found other research projects on mobile 3D that used these available technologies as basis for their test systems (Rakkolainen et al. 2000). The system components we used included the Virtual Reality Modeling Language (VRML) engine “Pocket Cortona” by Parallel Graphics, the Compaq iPAQ PDA running Microsoft CE 3.0 Operating System (OS) and various 3D design software.

Part of a larger system

We believe that mobile 3D is not a stand-alone technology, but holds the most promise when integrated with other forms of media in a true multimedia presentation. Our system centres on 3D because that is our focus, but there are many other applications where 3D would be a suitable enhancement. M3D will not likely be successful as an application in itself, but rather as part of a system designed for a purpose other than to display 3D graphics. For this reason we sought early on to develop our system within the framework of a multimedia standard, to allow for easy modification and expandability. Unfortunately, as shown in chapter 4, no true multimedia standards were available for handheld terminals at the time. As theorized by (Beck et al. 2002) there are indications that future handheld terminals will use the same system architecture as their desktop equivalents, making such problems obsolete. For the time being, however, we have to design M3D systems without adhering to a multimedia standard, but integrate non-3D system elements using bespoke solutions.

Wide selection of functions

We decided to give the users a wide range of options in viewing and navigating the world. We did not know how users would prefer to navigate the 3D world in a mobile setting. As explained in section 9.5, we suspected that the methods used to navigate 3D worlds on stationary systems would be less suited to a mobile setting. The users were also offered several different representations of the navigational data, including a walkthrough at ground level and a near-vertical overview of the entire 3D world. Giving the user several options was in part caused by our choice of a qualitative research method. We did not have a clear idea of exactly how users would respond to the system, so we gave them many different ways of performing the tasks and hoped to gain valuable input by observing how they reacted. To avoid confusion, we chose relatively few options that differed in significant ways from each other by being animated or stationary views, allowing different kinds of movement and perspective and so on.

1.5 Current research of M3D

We turned to established sources for general guidelines on system development. However, much of the research material this thesis builds upon is less known and applies to a very limited field within mobile IT use. Since we refer to many of these papers and studies on several occasions in this thesis they are collected and outlined below.

1.5.1 Modality theory of mobile IT use

In this thesis we often refer to the work by Kristoffersen & Ljungberg (Kristoffersen & Ljungberg 1998*a*). These authors are perhaps more well-known than the others that are referred to in this section. Almost all the other authors presented here refer to Kristoffersen & Ljungberg's papers on the methodologies of mobile IT-use. In their research papers, Kristoffersen & Ljungberg describe the *modalities* that they use to categorize different aspects of mobility. The concept of modalities is very far-reaching and covers many different aspects of mobility. This means that only a small part of this framework is applicable to our specialized area. Because their work is so well known, we use Kristoffersen & Ljungberg's model as the theoretical framework for mobile IT use. Thus we can place our area of research in relation to other mobile IT use and show similarities or differences.

1.5.2 Mobile 3D using VRML

The 3D City Info project (Rakkolainen et al. 2000) is very close to the one we describe in this thesis. It consists of a VRML model of the central parts of the city of Tampere, Finland. This model has been incorporated into a mobile system, allowing 3D images to be displayed on a PDA. Vainio et. al. have published a series of articles based on this project.

Another article that describes a VRML map on a PDA is “PDA Based Navigation System for a 3D Environment” (Brachtl et al. 2000). This paper deals with the possibility of presenting the information in a 3D form that provides the user with more depth and detail than information in a 2D form. A concept of such a navigation system is described together with all the partial problems that have to be solved. The result of the project is the implementation of a functional system that would be possible to use in various applications. This paper deals with the same problem area as us, but they focus on solving the technical issues rather than investigating such a system’s usability and/or usefulness.

1.5.3 Mobile 3D API

Games Application Programming Interface (GAPI) (Games Application Programming Interface 2003) is a set of 3D software development tools for Pocket PC handhelds. That mobile 3D is an area in rapid development is perhaps best illustrated by the fact that in the space of a year, GAPI went from being officially launched as a project to having two full games on the market and many more in development. As explained in chapter 4.7, the lack of a functioning 3D API has been a major stumbling block for developing 3D software on handheld devices. GAPI games and applications showcase the handheld terminal’s capability for 3D graphics with a quality similar to a stationary PC of a few years ago.

1.5.4 Indoor Navigation System

This paper(Butz et al. 2001) describes a hybrid building navigation system consisting of stationary information booths and a mobile communication infrastructure feeding small portable devices. The focus of the paper lies in resource-adaptive navigation systems and their underlying theories. Of special interest to us is the examination of the relationship between the accuracy of navigation data and the required level of detail of a graphical map.

1.5.5 Mobile computing in a fieldwork environment

Part of the "Mobile Computing in a Fieldwork Environment"-project (Pascoe & Ryan 1999), the paper “Using while moving” (Pascoe et al. 2000) is of special interest to us because it deals with how the problems of mobility and shared attention can be dealt with in interface design. Pascoe et. al. introduce two general principles in their interface design: Minimal Attention User Interfaces (MAUIs) and context awareness. These concepts are highly relevant to our investigation of shared attention in mobile IT. Also, Pascoe et. al.’s concept of fieldwork is similar to the use context of "true mobility". However, their goal for a MAUI is an interface that does not require the user to look at the PDA, and to operate it in one hand.

This rather radical approach makes the study less suited as a source of interface design ideas, but their concept of mobility is of great relevance.

1.5.6 Usability testing of mobile devices

In their "Speciale indenfor Human-Computer Interaction"(Beck et al. 2002) (approximately similar to a master's degree) Beck et. al. set out to create a theoretical framework for usability testing of mobile devices. Considering that "Metoder til brugbarhedstest af mobile apparater" is closer to a master's thesis than an academic paper, we do not build upon Beck .et. al.'s findings in the same way as the other articles presented here. Rather, we utilize their experiences in two ways: Primarily we use their methods for mobile usability testing as reference for our own experiment. In addition, they combine psychological theories of shared attention with the modalities of Kristoffersen & Ljungberg to create a conceptual framework for mobile usability testing. Though we follow a different procedure than Beck et. al. in many cases, this paper serves as an important point of reference since its scope is similar to ours and it was written so recently. Their work has been of great importance to us since it touches upon three areas of interest: Firstly their combination of cognitive psychology's theories of attention and Kristoffersen & Ljungberg's theories of mobility lies at the heart of our subject matter. Secondly they modify Kristoffersen & Ljungberg's theories with a focus on true mobility, making them more applicable to our case. Thirdly the end result of their thesis, the guidelines for conducting a mobile usability test are highly relevant to our own experiment design.

1.5.7 Augmented Reality

Though the system described in this thesis is not an Augmented Reality system by strict definition, AR had a central role as a model and inspiration throughout the project. We refer to many different AR studies in this thesis, primarily those that center on wearable examples e.g. (Reitmayr & Schmalstieg 2001) and (Hollerer et al. 1999). Common to all of them is that they point out the possibilities that lie in providing context-sensitive information that is linked to the user's surroundings. Though the AR approach is more direct and intuitive to users, it presents a very daunting set of technical challenges.

1.6 Paper overview

The outline of this paper is as follows:

In **chapter two** we describe the research methods we used to gain insight into M3D, the design of the 3D system and the execution and data-gathering of the experiment. The next chapter, **chapter three**, gives an introduction to 3D technology and Augmented Reality. We then describe the concept of Mobile IT use and introduce the concept of “true mobility” in **chapter four**. Technological constraints of the mobile terminal in light of M3D and a short description of the VRML standard are also included in this chapter. In **chapter five** we focus on Interaction design and describe metaphors and mental models as means of designing for interaction. GUI design in the form of dynamic user interfaces and principles of interaction design are also investigated. Shared attention is treated in **chapter six**, describing theories from the field of cognitive psychology and applying them to the concept of use in M3D systems. In the Design rationale chapter, **chapter seven**, we show the system requirements we used as the foundation of how our system coped with the shared attention problem. **chapter eight** describes the system in detail. **chapter nine** deals with the experiment design , and in **chapter ten** the results from the experiment are categorized and presented. The discussion chapter follows, in **chapter eleven**, collating the experiment findings and system development experience together with the theoretical foundation presented in the previous chapters. Finally, **chapter twelve** sums up our conclusions.

Chapter 2

Research method

THIS chapter gives an overview of the methods we used in our software design and subsequent experiment. Our choices of research methods and how we implemented them were in large part governed by our workflow.

2.1 Practical effectuation

The main inspiration for our thesis came from our observation that 3D use on mobile platforms was quite limited. To verify this and gather ideas for a thesis we conducted a pre-study of mobile technologies. During this pre-study we gathered material from existing literature on the subjects of mobility and RT3D and studied existing M3D systems. We also conducted interviews with experts in these and related fields. Much of this research was conducted in conjunction with our work for Octaga AS.

At the end of the pre-study we decided to test the usability and usefulness of an M3D system in a practical experiment. Unfortunately we could not find a ready-made system that we could use in such an experiment. The available systems differed greatly from our ideas of how an M3D system should be designed (see our "Design Guidelines" in the Introduction chapter for more on these). However, our pre-study of available technologies convinced us that it was practical for us to design our own system. We then formulated a set of system requirements detailing what such a system should be capable of.

The prototype design phase included finding a system model, designing a scenario and the actual system creation. To begin with, we explored several possible example systems that fulfilled the system requirements. Once we had decided on an example system, we designed a scenario for our experiment. This scenario mim-

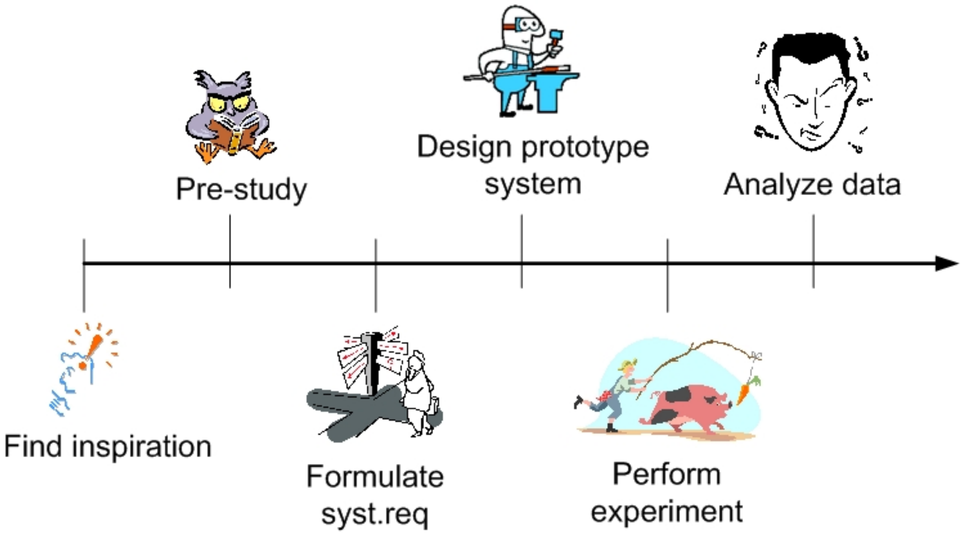

Figure 2.1: Sequential overview of our workflow.

icked situations where the system could be used. Based on the scenario demands we then charted out which parts of it would have to be fully implemented and which could be simulated. Our implementation strategy is covered in more detail below.

When the implementation was finished we performed a set of qualitative experiments on a small group of subjects. We aimed at collecting varied data from a scenario that was as close to actual use as possible. After the experiments we started systemizing and analyzing the data we had gathered.

2.2 Qualitative Research

A common denominator for our research methods was that we used a *qualitative* approach to the subject matter. Where *quantitative* research relies on large samples and numerical analysis, a qualitative research strategy relies on in-depth studies of a smaller sample (Silverman 2001).

The most basic reason for this choice of approach can be found in the nature of our thesis. We wished to explore the potential for mobile 3D systems. This meant we were more interested in *how* users would utilize an M3D system than *how well* they would utilize it. The last question could perhaps be better answered with running experiments on a large group of subjects and time their progress. We found that a qualitative method was better suited to determine how the subjects reacted to and made use of this new tool. If the optimal way to utilize an M3D system had been known to us, the study would have been more comparative and detailed. This would have made a quantitative approach more suitable. In-depth observations and interviews were necessary to determine how and why they did as they did.

> *"Qualitative methods are content addressing, while quantitative methods are more guided by their content."*

- Translated from (Wideberg 2001).

Qualitative research is often linked to an *explorative* approach to the subject matter. Using this method, the researcher uses several different techniques to obtain information about a certain problem area. This method is often used when prior knowledge about the subject matter is limited. We found that this method was well suited to our needs as we had little prior knowledge of how mobile 3D systems could be used.

The choice of research method is often linked to the available resources (Skog 1998), which was also the case for us. We had just two researchers and one PDA available, so performing enough tests to provide valid statistics would have been impractical.

2.3 System Development Method

We used a rapid, incremental approach similar to the "quick and dirty evaluation" method of (Preece et al. 2002, 341) when creating our prototype system. In essence, we produced an early, incomplete version of the system, tested it ourselves and on fellow students and incorporated the feedback into the next version. These activities were repeated several times and on different scales as we made large and small modifications to the system. On two occasions we demonstrated unfinished versions of the system to RT3D professionals at Octaga and received valuable feedback. The main benefit of this method was that we could produce a working system very rapidly. This was essential to us since we had to weigh many issues against each other to create a system that worked within the limitations set by the mobile terminal. Tweaking each feature to create a balanced system was quite time-consuming and required us to explore many different approaches to each feature. A major consideration for us was the speed or *frame rate* of the system. A good frame rate was essential to convey motion and make the 3D world come to life. To achieve this we spent a lot of time optimizing every feature and minimizing the detail of textures and models. We also considered functions such as Level of Detail (LoD) to see if we could achieve better frame rates through them. A more structured approach would have made this last stage easier, as we could have implemented cost-efficient features earlier. But this would have meant that the initial design stages would have taken more planning and thus more time. We also had to learn how to program in VRML and ECMA-script, which neither of us had much experience with. The rapid but unstructured approach we chose allowed us to experiment on and learn from the system itself.

2.4 Experiment Method

We wished to examine both the *usability* and the *usefulness* of the system. A laboratory experiment would probably be of little use to investigate the usefulness of the system, as such a setting would differ greatly from daily life. It would also be difficult to recreate mobility in a laboratory. A hybrid approach of a laboratory experiment and a field study, such as the one used by (Beck et al. 2002) held more promise. We decided to test the system in a setting where it would likely be used, but we retained control of what tasks the subjects should perform and recorded their actions throughout the experiment. One method, *contextual inquiry*, described in (Beyer & Holtzblatt 1998) seemed to suite us well.

2.4.1 Contextual inquiry

Contextual Inquiry (CI) is based on ethnography and sociological research tradition where the researcher/observer goes into the research object's own environment (Beyer & Holtzblatt 1998). The researchers observe the potential users of the developed product for a period of time, typically a few hours. The observer stays in the background for most of the time, but also inquires about events that are not obvious but may be significant regarding the focus of the research (Väänänen-Vainio-Mattila & Ruuska 1998). CI is an explicit step for understanding who the customers really are and their daily routine. (Beyer & Holtzblatt 1999). A contextual interviewer observes users as they work and asks about the users' actions step by step to understand their motivations and strategy. In addition to notes from the observations and inquiries, work products such as data sheets, notes and other artefacts from the environment can be collected. These artifacts are collected for later reference about the user's specific tasks and work practices (Väänänen-Vainio-Mattila & Ruuska 1998). Väänänen-Vainio-Mattila and Ruuska (1998) report that CI is successful in the use of developing mobile communication units at Nokia. Our CI has not been concerned with the development of a service, but rather how 3D affects the problem of shared attention.

The experiment we designed consisted of making a mobile 3D navigation system available to the users and giving the subjects a series of tasks to perform. Most of these tasks consisted of locating some building or feature on the campus. We did not instruct the subjects on how to “best” use the system or how to perform each task most efficiently as we were primarily interested in seeing whether or how they made use of this new tool. We used several techniques to gather data on how the subjects interacted with the system and their environment and from this we hoped to gain some insights into the potential of M3D systems.

Time and resource considerations did not allow us to use a control group. It would also be difficult to outfit this control group with ”traditional” navigation tools to ensure relevance. Our system has no direct counterpart in 2D maps, both because the system offers more than just a 3D version of a traditional map and because 2D maps are often used in concert with other tools such as a guidebook or a compass. Instead we chose to compare the subjects’ performances to how a person familiar with the campus fared. From this we hoped to determine if using the system demanded so much time and attention as to be impractical.

2.4.2 Data collection

Our literature recommended using several different means of data collection (Preece et al. 2002, page 349). Each technique yield a different kind of information, slanted from a certain perspective. Using different techniques ensures that the data comes from different perspecives. If several techniques yields similar findings, this corroborates the find. Also, our sample group of test subjects was quite small, so relying on a single technique meant risking not getting sufficient data. The nature of the experiment posed restrictions to certain techniques (such as for observation explained below) so we sought to supplement these techniques with other forms of data collection. The three methods we used were: direct observation, thinking aloud and user interviews. Each is explained in more detail below.

Observation

The nature of the experiment required that there was a researcher near the subject to help with technical difficulties and deal with unforeseen circumstances. We decided to have this test person monitor the subjects’ progress and take notes. An important issue would be the degree of invasiveness of the observers. On the one hand we wanted to acquire detailed and accurate data, requiring the observers to closely follow the subject, but on the other hand we did not want the subjects to become too self-conscious or nervous because of this. We decided to have a single researcher follow the subject around during most of the experiment. The subjects were explained that the researcher would take notes but would primarily be there to offer technical assistance and make sure nothing untoward happened. We went to some length to make the subjects feel comfortable during the experiment, as we tried to make the setting as close to normal social interaction as possible. Since there was a possibility that having a researcher following them closely during the tasks would affect the subjects’ performance, we arranged that they would complete the first task apparently unsupervised. For the first task the subjects would be given the PDA, told to complete the task displayed and that a researcher would meet them at their destination. The researcher was in fact located where he could observe the subjects easily without being seen himself. Hopefully we could determine whether the subjects acted differently while supervised than when apparently unsupervised. This could also make the subjects less aware of the experiment context by accustoming them to concentrate on the PDA rather than the researcher.

As a final incentive for us to conduct direct observations of the subjects' behaviors we wanted to hold this data up against the other two data gathering techniques, namely interviews and thinking-aloud. These techniques focused primarily on the *users'* interpretation of their actions and could be slanted by their perspective. Observation would provide us with another perspective on their actions that was perhaps more objective. According to (Koenen 1993) what the subjects actually did should be weighted more heavily than what they *said* they did.

Interviews

We planned on conducting interviews of the subjects immediately after they completed the experiment so that they would have this fresh in their memories. According to (Preece et al. 2002, page 390) there are four kinds of interviews: *open-ended*, *structured*, *semi-structured* and *group*. Of these, the first three were applicable to our research. An open-ended interview takes its direction from the subject and follows where he[1] wants to go in describing his impressions. Structured interviews in contrast are wholly controlled by the researcher who follows a predetermined list of questions, while semi-structured interviews blend features from both styles. Choice of interview style should be governed by the subject matter at hand (ibid.). Since our goal was to gain impressions and feedback on a new concept, we opted for an informal, open-ended interview. But as in any interview we had to balance between passivity and over-direction (Myers & Avison 2002). To make sure we had a starting point and to provide a similar outline to the interviews we made use of an interview guide. This took the form of a list of questions that was intended as a guideline for discussion rather than a strict template and can be found in Appendix C. The interview guide also helped us to prompt the user if he felt he had nothing to say of his own accord.

This choice of interview style reflected our qualitative approach and is in many ways typical of qualitative research interviews (Silverman 2001, page 26). Qualitative researchers often rely heavily on in-depth interviews of subjects to determine their reasons for acting the way they do. However we were wary of putting to much weight on our interview findings as this could be seen as adopting the sub-

---

[1] For readability purposes, this thesis uses standard masculine pronouns when referring to persons of uncertain gender. In such cases, these pronouns are intended to convey the meanings: he/she, her/his, etc.

jects' point of view as an explanation. This was the primary reason for including other data gathering techniques to supplement the interview data.

Thinking aloud

The thinking aloud protocol is often used in usability studies. In essence, researchers provide the test users with the product to be tested (or a prototype of its interface) and a set of tasks to perform. They then ask the test users to perform the tasks using the product, and explain what they're thinking about while working with the product's interface. The advantage of using this protocol is that a lot of qualitative data can be collected from only a small number of users. Since the user thinks aloud while interacting, the experimenter gets a very direct understanding of what parts of the dialogue that caused the most problems. Additionally, problems that the user would not remember in an ordinary interview might show in a thinking aloud session. In our experiment, we decided to implement the *Simplified Thinking Aloud* protocol described by (Nielsen 1994). A full TA study usually requires trained psychologists or user interface experts. The Simplified Thinking Aloud method allows us to perform these roles ourselves. It also allows the method to be used outside of a laboratory setting.

We hoped that TA would allow us to catch information on the users' internal processes that we might miss in direct observation. This could allow us to gauge where the subjects directed their attention. Though the nature of the information is similar to what we asked for during the interviews, the users might provide interesting information this way that we did not think to ask for. The users might also just forget to mention this information during interviews, even though the interviews were conducted immediately after the experiment.

Chapter 3

3D- and AR technology

IN this chapter we introduce the technology of 3D computer graphics and Augmented Reality (AR). Our M3D system is inspired by AR and uses 3D technology as basics for interaction and visualization.

3.1 3D technology

Intuitively, speaking of three dimensional computer graphics is a bit of an oxymoron, as the computers today rely almost exclusively on 2D screens to display images. 3D visualization methods are still in their infancy, and are outside the scope of this paper. By 3D computer graphics we here refer to virtual objects and environments that can be displayed from any viewpoint. Furthermore, 3D data commonly take two distinct forms: real-time and pre-rendered. A RT3D application is capable of displaying and manipulating three-dimensional objects, most commonly in response to user demands, almost instantaneously. Examples of software that use this technology include CAD/CAM software and computer games and simulations. Real-time is very demanding on the computer hardware as it requires the computer to maintain all the objects to be manipulated in working memory. In contrast, when computer-generated 3D objects are displayed in the same way as a 2D movie, it is referred to as pre-rendered. Pre-rendered 3D is very popular in movies, with scarcely any major production without a substantial computer graphics budget. In this paper we primarily discuss RT3D applications.

The methods and formats for representing 3D objects today are centred on the concepts of polygons and textures. There are other ways of representing 3D data, but with the current limitations of computers and the widespread use of 3D Application Programming Interface (API's) like OpenGL (Open GL - www.opengl.org 2002) and DirectX (DirectX 2002), this method seems to remain the standard in the foreseeable future. In this chapter we will describe some of the key concepts that lie behind RT3D, their history and what they mean for mobile 3D application developers.

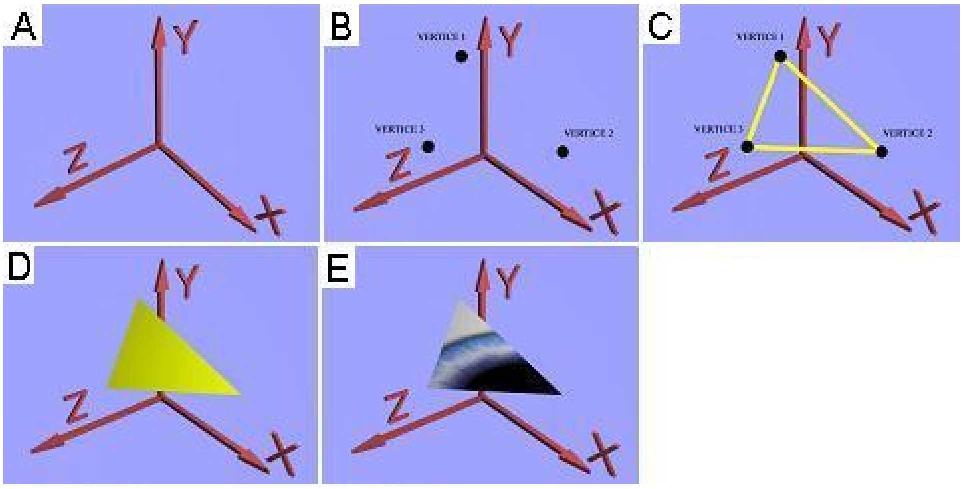

Figure 3.1: 3D basics: A) The Cartesian coordinate system with 3 axis. B) Three points, or vertices, placed in the coordinate system. C) The three vertices are connected by three straight lines: edges. D) The three edges encompass a triangular area called a face. Note that the vertices and edges are invisible in this rendered image, as will mostly be the case. E) Here the triangle has been given a basic texture. It will later form a part of an eye. (Illustration by the authors)

3D space

In order to simulate 3D objects, the computer requires a mathematical model of three-dimensional spatial relationships. The most common way of describing and delineating 3D space is by a Cartesian coordinate system (Foley et al. 1994, page 59) (figure 3.1 A). This system consists of 3 vectors originating at a point called origo. These vectors stand at right angles to each other and are labelled by long-time convention as X, Y and Z. This is a very efficient way to describe 3D space, so it is used by almost all RT3D systems.

Vertices

The computer represents a location in 3D space as a set of three numbers. These numbers give the locations along the X, Y and Z-axis of the coordinate system respectively. Points in 3D space that are located on the surface of an object are called vertices (singular: vertex) (figure 3.1 B). Vertices have no volume (they are "zero-dimensional") and so can not be seen in the final image.

Edges

Edges, like points, are invisible in the final image, since they have no volume. In RT3D applications, edges are usually completely straight as this requires the least computational power (figure 3.1 C). In pre-rendered 3D and in some very advanced real-time systems edges can be mathematically modelled to be curved. Examples of such curved edges include Bezier curves and B-splines. In these cases, the vertices are called control points, and are not necessarily located on the curve. These techniques are very rarely used in RT3D.

Triangles

Three edges that are connected to each other by three vertices form a triangle in 3D space (figure 3.1 D). The triangle is the most basic 3D shape that encompasses a surface. In most 3D applications, all objects consist of such triangles. Sometimes objects are depicted as consisting of squares instead of triangles. These squares in reality consist of two triangles that share an edge. Both triangles and squares are referred to as polygons. A very important characteristic of any 3D object is the number of triangles or squares that it consists of. This number is often called the polygonal count, or polycount for short, of the model. The name polycount is used regardless of whether it refers to triangles or squares. In this paper we will give the polygonal count, when it is necessary, as the number of triangles.

The Mesh

When many polygons are joined together to form a 3D object, the vertices are usually located along the surface of the object. When all the vertices are arranged in this way the result is an empty "shell" that forms the borders of the object (figure 3.2). 3D meshes are usually closed, so even though it consists solely of a 2D skin folded into 3D, the object will appear to be solid. Also, since most objects are opaque, there is no need to model their interiors.

Texture

The area between the three vertices that make up a triangle is called its *surface*. The surface is most often single-sided, meaning that it is transparent from one side, but opaque from the other. What the opaque side looks like is determined by its texture (figure 3.1 E and 3.2). The texture is usually an image such as a photograph. This image is most often stored in the computer as a separate file and "wrapped" around the 3D object when the object is viewed by the user. Each polygon is given a section of the image, which is overlaid on it. The 3D graphics software then calculates how the section will look if viewed from the same angle as the polygon. By repeating this process with all the polygons of the object, the object appears to have a "skin" based on the original image. The texture can have many qualities: colour, roughness/smoothness, shininess, reflectivity, etc. that we will not go into

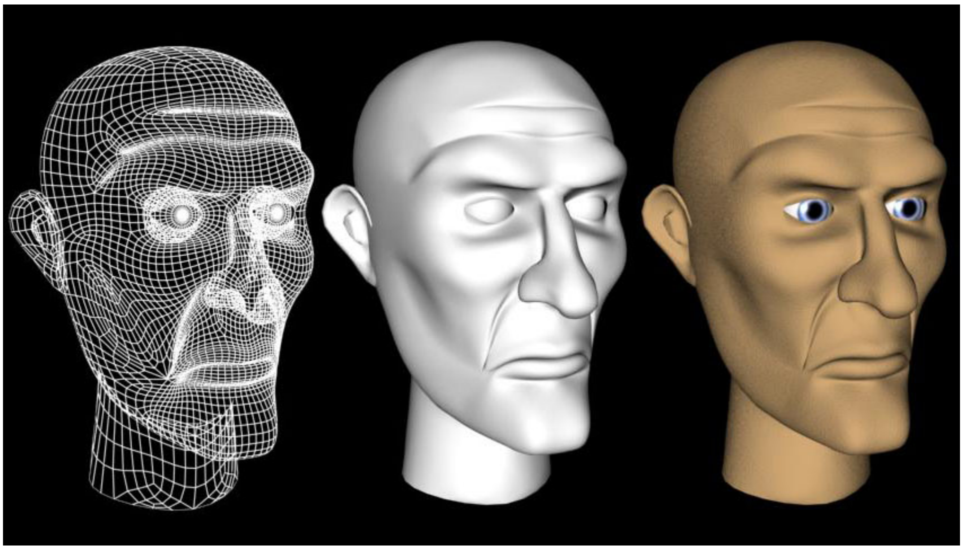

Figure 3.2: This is an example of a relatively complex 3D model. At first the mesh is formed by polygons, then a surface is applied, and finally a texture image is wrapped around the object. (Illustration by the authors)

here. In most RT3D applications, a simple colour map is sufficient. This method of building objects is almost universally used in today's RT3D applications. Even the most complex shapes can be created using polygons with textures.

Displaying the image

The process of displaying 3D objects on the user's 2D screen can be divided into three sections: geometry manipulation, rendering and lighting. Geometry manipulation is the process of moving and changing the 3D meshes in accordance with the application's and the user's demands. During rendering the 3D graphics engine translates the 3D data into a 2D image by applying perspective in a way that mimics reality. The lighting stage applies shades, reflections and other effects to the rendered image. In RT3D the shader algorithm is responsible for smoothing the light levels across an object, so that its surface is not broken up by the polygon edges. The shader only works on the surface of the model, so the outline will still appear jagged if the object consists of few polygons. If each triangle has separate light levels (which is what would appear in nature) the polygons are said to be flat-shaded (figure 3.4 A). When the shader blends the light levels across the triangles of a mesh, the polygons are said to be smoothshaded (figure 3.4 B).

The Graphics engine

The piece of software that handles the 3D graphics is commonly called the "graphics engine" or just the engine. This software can be shared by many applications,

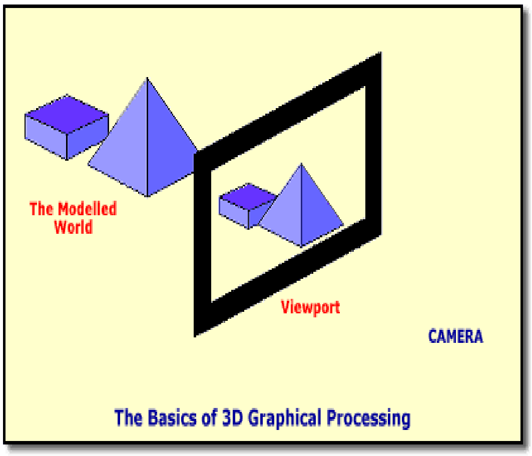

Figure 3.3: The 3D modelled world is translated and rendered to 2D (viewport). The camera illustrates a user. Illustration taken from: (PCWorld.no 2002).

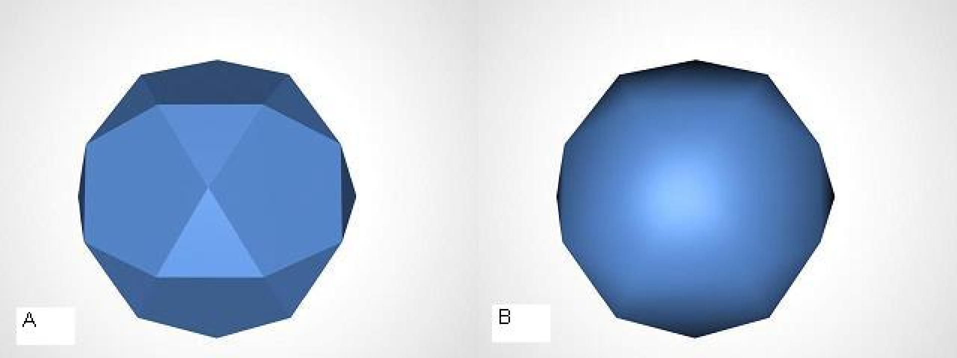

Figure 3.4: A) Here is a flat-shaded dodekahedron (12-sided polygon). Note that each of the twelve sides are in a uniform shade. This is because each side consists of three triangles that are parallel and so reflects the same amount of light. B) Here is the same dodekahedron with a smooth shader applied. Note that while the outline has not changed, the interior looks like a sphere.

and can display any content that is within its capabilities. Both computer games and commercial 3D standards contain 3D engines. Today, most 3D engines make use of specialized 3D graphics API's like OpenGL or Microsoft's DirectX. A detailed analysis of these API's are beyond the scope of this article, but it suffices to say that they greatly enhance RT3D applications by supplying a set of primitive 3D shapes that can be swiftly and easily manipulated.

Other methods

There are some other techniques to produce 3D computer graphics, and a few have played a role in mobile 3D applications. A notable example is the voxel engine. Rather than using vertices and planes to delineate an object, such an engine would build all objects using three dimensional volume cells called "voxels" (for VOlume piXEL). These cells most often take the form of cubes, or sometimes 2D squares. The advantages of the voxel engine lies in its ability to model complex shapes without increasing computer workload and its ability to create "solids" by filling objects with more voxels. Unfortunately, voxel worlds do not scale well, since all the voxels in a world have the same dimensions. Thus a big object would require a lot of computing power to render. Still, voxel engines have found a niche in medicine, where they are utilized because of their ability to model the interiors of objects. See for instance (VOXEL-MAN 3D-Navigator: Brain and Skull 2002). Also, voxel engines have characteristics that are different enough from polygonal engines to be superior in special cases. An interesting example is the game Chopper Alley by Amazing Games, one of the first 3D games on a handheld computer (figure 3.5). In this case, a combination of voxel and polygonal engine made for the best result on the handheld's particular hardware.

3.1.1 A brief history of 3D technology

RT3D is so common today that many household computers sport specialised hardware in the form of 3D graphics accelerator adaptors. These “cards” are designed to take some or most of the workload off the Central Processing Unit (CPU) when displaying RT3D objects. Most current computer games are completely reliant on a fast graphics card in order to run properly. The idea of creating specialised hardware for RT3D applications is not new. In the early 1980s, Jim Clark had a vision of building a 3D graphics engine on a chip. Clark, along with several of his students, took this idea and formed Silicon Graphics, where the first 3D graphics workstation was developed (Baum 1998*b*). In 1984, the Silicon Graphics IRIS 1400 integrated a workstation with 3D hardware support. The graphics hardware of this and similar first-generation systems were optimised for a single application like a flight simulator and exclusively rendered flat-shaded polygons. The first example of a second generation system was the Hewlett-Packard SRX, followed by the Silicon Graphics GT, which was able to render over 100.000 triangles per second and provided facilities for smooth-rendered polygons (Baum 1998*a*). Third generation systems and

Figure 3.5: Screenshot from Amazing Games' Chopper Alley. Note the smooth contours of the landscape, accomplished through voxels.

beyond are truly general-purpose and can be used for a wide variety of programs. Today's 3D cards are primarily rendering accelerators. That is, the 3D graphics card handles all the chores of converting the calculated geometry to an image that can be displayed on the computer screen. But much of the workload of moving and manipulating the 3D objects is still carried by the CPU. Future graphics cards will probably take over more of the geometry manipulation and lighting tasks, resulting in more complex geometries and more realistic lighting effects.

The importance of games

It is impossible to compile a history of RT3D without mentioning computer games. 3D computer games have driven the rapid development of 3D software techniques and hardware (ibid.). The very first computer games to approximate 3D used what is called vector graphics. An excellent example of this type of game is "Zoids" from 1986. 3D vector graphics create an environment that consists solely of straight lines, most often in black and white. There are no planes or textures, and the geometries displayed are very simple. The game most often credited with being the first "3D game" is Wolfenstein 3D by Id software, released in 1992 (White 2002) (figure 3.6). This game created a very simple illusion of 3D motion. However, it was a huge breakthrough in that it displayed rudimentary colour textures and introduced the "first person shooter" game format. The various follow-ups to Wolfenstein included the lauded Doom, also by Id software, as well as a myriad others.

Figure 3.6: A screenshot from Wolfenstein 3D. Note that there is no texture on the floor or ceiling. Only the walls are 3D objects.

However, in terms of 3D graphics, the next revolution came in the form of Quake in 1996 (White 2002). The game Quake (also by Id) was the first game to feature a true 3D world, where everything was modelled using the polygon technique (ibid.). The previous games had all used 2D graphics within the game to reduce computer workload and make the game run faster. Quake was built on advanced algorithms that allowed for the necessary speed to create a true RT3D experience. It is especially in this regard that games have provided the biggest contribution to RT3D. Computer games must run very fast indeed, so a lot of work is put into optimising the code. Unfortunately this produces several drawbacks: The demand for speed is so great that game programmers are often forced to take mathematical "shortcuts". These often take the form of approximations and result in a less accurate simulation. Often they also produce side effects such as "aliasing", where objects suddenly take on jagged edges. These drawbacks mean that game engines are seldom used for more "serious" 3D simulations such as architecture programs. There are notable exceptions, though, where a game has crossed the gap to simulation. Most notably "Marine Doom", a Doom II modification used by the U.S. Marines to train soldiers during the 80's (Marine Doom 2002). Because they are so specialised, 3D games are capable of dazzling graphics that put most public 3D standards to shame (Baum 1998*a*).

3.1.2 Consequences for the mobile computer

This chapter has recapped some of the important factors and concepts used in the development of RT3D. The current methods used in stationary 3D applications and their history is bound to have a profound effect on the development of mobile 3D. Mobile 3D systems will probably use the same techniques as their stationary counterparts and will maybe even be modified versions (or "ports") of the same systems. The issues that shape current 3D graphics engines, such as use of standard API's, are just as relevant for mobile systems. Still, mobile platforms are different

Figure 3.7: A screenshot from Id's upcoming Doom 3. This represents the improvements in RT3D graphics that have occurred in just 10 years.

enough from desktop workstations that special solutions may be found to produce better results. It is important to be aware of the past history of RT3D when designing new system: When one takes into consideration the specifications of handheld computers today and the software available to developers, the parallel to the situation on stationary systems is immediately apparent. Solutions that were applicable to stationary systems then, but have since been discarded can be very relevant to mobile developers today. 3D graphics hardware support has proven crucial to 3D applications on stationary systems and this may prove to be the case for mobile devices as well. Games have been integral to the development of RT3D software and hardware in the past, and with the rapid growth of the games industry we may well see the same effect on mobile platforms.

3.1.3 Choosing 3D as visualization method

The popularity of 3D technologies has increased rapidly during the past few years (Baum 1998*a*) 3D graphics takes the traditional text and window displaying method to a new level by converting objects from the real world into digitized versions or metaphors to present for a user. Why 3D in some fields is preferred over 2D can be related to how humans interpret graphical visualization and how 3D evolves an immersive experience for a user. Digitized 3D worlds is an approximation or a substitute for the real world or an imaginary world. The strength in this is that we already know how to operate in the real world, and can readily transfer this knowledge to the 3D world. Digitized 3D objects are recognised quickly because we already have an understanding of the object by real life experience. The challenge however is to make the digitized version realistic enough - not just graphically but also with respect to interactions with the rest of the 3D world. It is not enough for an object to appear realistic, but it must also react to stimuli in a way that the user expects it to from real world experience: A drawer can be opened by interacting with its knob, a heavy object falls to the ground differently than a feather etc.

A commonly accepted notion in 3D virtual worlds is that of immersion. As pointed out in the article (Holloway 1995): *"the basic idea with virtual reality is to immerse a user inside an imaginary, computer-generated "virtual world""*. This same effect also holds true, though in a lesser degree, for 3D worlds in general. Basically, the idea of immersion is that the user is completely “submersed” in the virtual world to the point of forgetting his physical surroundings. This effect is obviously detrimental to our goal of a mobile 3D system being operated concurrently with real-world tasks. However, in this project, we take a contradictory point into consideration: namely the inherent realism of 3D worlds. With realism we mean effects such as perspective, dynamic movement, interaction with objects and so on. We will argue that it is possible to achieve recognition through realism. One of our major points of interest is whether this effect is able to “cancel out” or compensate for the effect of immersion.

3.1.4 Recognition of 3D objects

In designing a highly visual system, we found a need to investigate humans’ ability to interpret and understand complex graphics layout. This is of particular interest to us because object *recognition* is so important in this system.

The idea of quick and easy recognition of real world objects from 3D representations is central to our rationale. The connection between computer-supplied information and the real world rests on the user’s ability to quickly connect a 3D model with its real world counterpart. The fact that our surroundings are three-dimensional opens the possibility that a 3D map should be more intuitive than a 2D one. But this depends on the human mind gaining some advantage from the 3D representation that is not there in 2D images. The most obvious advantage comes from the 3D data itself. A 3D model contains spatial information that is simply not there in a 2D map. And because of our familiarity with 3D objects from real life, 3D computer generated objects should logically be easier to recognize. But this hypothesis relies on the human mind being able to process 3D data directly.

Directly recognizing 3D objects by matching them against reconstructed 3D data is not necessarily the method used in human vision. Instead, the process of perceptual organization, which detects parts of objects that are likely to remain invariant over wide ranges of viewpoints plays an important role in human visual recognition. But this process is not infallible, so it is used mostly as a primary stage of recognition that triggers a viewpoint-dependent analysis. A quantitative method is used to simultaneously determine the best viewpoint and object parameter values for fitting the projection of a 3D model to given 2D features (Lowe 1987).

Figure 3.8: Illustration: A model posing as a subject is comparing the virtual world presented on the iPAQ with her environment. The respective viewpoint from the 3D world is shown inlaid. Note that the actual experiments were performed during summer.

Even though this model has gained some acceptance, the degree in which 3D structural information is used in object recognition remains an area of strong debate (Liu & Kersten 1998). Though the degree in which people use 3D data in their mental processes is unclear, 3D models provide functionalities that mimic the real world such as perspective and variable viewpoints. Even if the human mind does not use 3D data directly, it can use these secondary characteristics for recognition purposes. One of the main advantages of a 3D system for recognizing objects is that it can reproduce any viewpoint imaginable. This advantage is independent of the exact role 3D data plays in the human cognitive processes.

This myriad of possible viewpoints raises an interesting problem: 3D objects may look different from different angles. This knowledge is so ingrained in the human psyche that the expression *point of view* is used as a metaphor in such different languages as English, Hebrew and Russian. This could mean that to recognize a real world object from a 3D map requires that the user is capable of recreating the exact perspective that he is experiencing in the real world. This would place great importance on the usability of the navigation interface of a 3D map. But the degree in which recognition is viewpoint dependent is debated (Liu & Kersten 1998). Obviously, having a similar viewpoint as reference is advantageous, but recognizing a real world scene from a 3D representation does not necessarily require exact precision. The human visual system exhibits an impressive ability to recognize objects when viewed from a different perspective (Edelman & Weinshall 1994). This ability allows a 3D map system to supply useful information to the user without

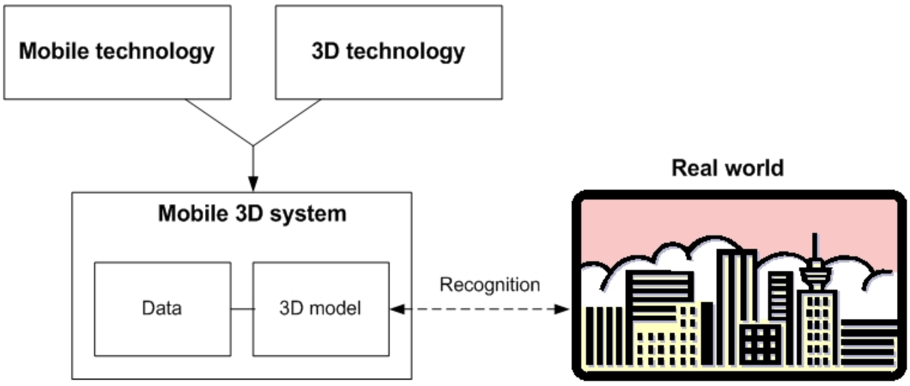

Figure 3.9: Illustration of how our system is derived by combining mobile technology and 3D technology and how the 3D model relates to the real world by recognition.

requiring him to grasp and manipulate the 3D data directly.

In chapter 7 we derive a system requirement where human perceptual skills in fast recognition is a central part. This system requirement deals with the use of realism to link the 3D model to the actual environment.

3.2 Augmented Reality

Augmented Reality (AR) is a variation of Virtual Reality (VR). But while VR immerses the user completely in an artificial world, AR allows the user to see the real world, with virtual objects superimposed upon or composited with the real world (Azuma 1995). These virtual objects often supply information about real objects and so are often referred to as annotations. The vision of AR is a world where a user is experiencing a blend of physical and virtual reality simultaneously, interacting with both equally naturally. Empirical evidence suggests that annotations registered to real-world scenes features offer benefits for communicating task details (Neumann & Majoros 2002). Why is combining real and virtual objects in 3-D useful? According to researchers, AR offers intuitive and natural means for people to navigate and work effectively in the real world (You 2002). AR enhances a user's perception of and interaction with his surroundings. The information conveyed by the virtual objects can help a user perform real-world tasks as the virtual objects display information that the user cannot directly detect with his own senses (Azuma 1995). AR operates successfully in some areas today and remains an area of much interest and ongoing research. While some researchers define AR in a way that requires the use of Head-Mounted Displays (HMDs), Azuma defines AR as a system that have all of the following three characteristics:

1. Combines real and virtual elements

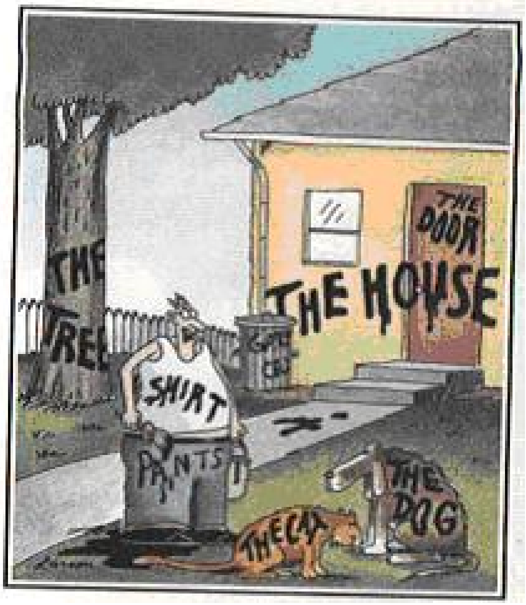

Figure 3.10: "Now! ... That should clear up a few things around here!". Taken from “The Far Side” by Gary Larson.

2. Interactive in real time
3. Registered in 3-D

This definition allows for other technologies besides HMDs while retaining the essential components of AR. It excludes film or 2-D overlays. Today’s Hollywood films feature photorealistic virtual objects seamlessly blended with a real environment in 3-D, but they are not interactive. 2-D virtual overlays on top of live video can be done at interactive rates, but the overlays are not combined with the real world in 3-D. However, this definition does allow monitor-based interfaces, monocular systems, see-through HMDs, and various other combining technologies. Note that Azuma finds that 3D is an integral element of AR, but that mobility is not. The two concepts are interrelated in that it is necessary to record the user’s motions, if only his head movements, for any type of AR system to work. There are many examples of AR systems that are not mobile in the sense of being wearable (Schnädelbach et al. 2002), but they invariably require that changes in the user’s viewpoint are recorded or controlled. Even using this definition, AR is not as of yet adopted by the society, but rather restricted to a few highly specialised and high-tech areas such as fighter pilot navigational aids or they are experimental prototypes like robot navigation systems. The main reason for this lies in the technological challenges that need to be overcome before one can create a truly mobile AR system. These challenges are mainly concerned with making the systems mobile by making them small, lightweight, giving them sufficient power, and so

Figure 3.11: Illustration: An example of what an AR system might look like from the movie Terminator 2. The hero sees enhanced outlines and technical information overlaid on his field of vision. Here he is looking at a motorcycle.

on; creating functional HMD's; but perhaps most importantly: accurate tracking (Azuma 1999). It is important to note that the characteristics of AR systems above apply to any kind of AR system, but the technological challenges listed are particular to truly mobile AR systems.

The problem of accurate tracking is pervasive and not specific to AR systems: There is a wide range of applications that could take advantage of such technology. Accurate tracking has been the subject of a great deal of research, but position and orientation tracking is *"one of the areas that has seen insufficient innovation in the past decade"* (Zyda & Sheehan 1997).

3.2.1 The AR metaphor

So is what we're doing really designing a "lightweight" AR system? The answer to this question is no. Our reliance on a representation of reality that co-exists in the user's field of vision means that our system cannot be considered augmented "reality". But, we do aim to offer context-sensitive information to users and link this information to objects in the world around them. So our system and AR have similar goals, but we do not mean to make an "ersatz" AR system by any means. Rather we use the model of augmented reality for inspiration and guidance, since our goals are so similar. We have chosen the PDA as our platform, and we have to make our system suitable for this type of terminal. Our method for location detection capitalises on the available system components, but is by no means fail-safe. Perhaps the most important weakness of our proposed solution is that it relies on the user being able to accurately locate his own position in the 3D world. The user must accomplish this task through the process of recognition: accurately connecting the 3D model to reality in his mind. We see the problem of facilitating recognition as part of the challenge of creating a mobile 3D system that aids the user while he is on the move. The user must in effect share his attention between the system and the real world. Quick and easy recognition of real world objects is essential if the user is to exploit the information the system provides. In chapter 6 we discuss this problem further and outline our strategies for facilitating recognition and compensating for shared attention.

Chapter 4

Mobility

MOBILE technology and mobile services are becoming more and more commonly available. There is still a push for new consumer markeds, or a comsumer demand for new services, if you will. It is believed that the mobile comsumer marked is far from sated as of yet, and telecom companies are frequently exploring new fields to expand their markeds. Technology and services in mobile IT use are often closely connected. As opposed to stationary IT use where the work situation is local, i.e the environment do not change, a mobile situation requires services to take the environment into account.

In this chapter we will take a look at what mobile informatics means and what is expected of it. There are several difficulties to be mentioned and a diversity of fields of mobile informatics that address these difficulties.

4.1 True Mobility

The meaning of mobility is often used as a denomination for radically different use modes and contexts. This thesis is primarily concerned with what we call "truly mobile computing" or simply "true mobility". That is; using a computing device while at the same time being in motion. "Being in motion" refers primarily to walking, though bicycling or driving a car are two other examples. In this thesis, true mobility is put in contrast with "portable" mobility, which is that a user moves between locations but sits down and uses the device as a very small desktop computer.

There are many researchers who have investigated use patterns that could be called examples of true mobility. But these researchers typically use their own terms and vocabularies, and there is little consistency. A few of the most notable examples are briefly outlined below.

A variety of different studies (Reitmayr & Schmalstieg 2001) , (Feiner et al. 1997), (Allison et al. 2000), describe mobile Augmented Reality systems. The concept of mobile computing in these studies has many similarities to our "true mobility". The systems are designed to directly composite computer-generated images in the user's field of vision. These images typically take the form of supplementary information linked to real world objects, called *annotations*. This is very similar to the idea behind the system described in this thesis. Because of this similarity, the concept of "mobility" in these studies is almost identical to ours.
However, in these studies, a mobile AR system must be man-portable and designed for use while in motion. It is therefore not necessary to contrast the term "mobility" with "portability" or "stationary use". In this thesis, the AR concept of mobility is used only as an inspiration, or an example of a mobile use-pattern.

In "Using While Moving: HCI Issues in Fieldwork Environments" Pascoe et. al. are concentrating on developing services for scientific field workers. They stress the rigours that face the fieldworking PDA user: crawling, squatting and running. All while using a PDA. Though daily life puts less extreme demands on a PDA user, Pascoe et. al.'s concept of fieldwork fits remarkably well with what is here referred to as true mobility. Pascoe et. al. refer to this use pattern simply as "mobile computing" to contrast with "portable computing" or "static use". Though our concepts are similar, Pascoe et. al.'s term "mobility" denotes a very specific branch of users (fieldworkers) and is difficult to contrast verbally to other types of mobility.

Kristoffersen & Ljungberg build their reference model from a very basic definition: "Mobile IT use is a mobile person's use of IT." In their mobile informatics reference model, the modality that closest resembles the concept of true mobility is called “Wandering”. Kristoffersen & Ljungberg define it as "working while being locally mobile" (Kristoffersen & Ljungberg 1998*a*). However, this concept is too broad to be of much use to a study on true mobility use. Both (Pascoe & Ryan 1999) and (Beck et al. 2002) point out that the concept of wandering focuses on the mobility of the *user* and the *device*, but not the *activity*. For a more detailed explanation of K&L's modalities, see the next section of this chapter "Describing Mobility by Modalities".

In their article "Spatial Cognition and Natural Language Interfaces in Mobile Personal Assistants" Kray and Porzel speak of "Truly mobile systems". These are PDA programs that are designed to be used while on the move. This is precisely the concept of mobility that is used in this thesis. Kray and Porzel have a different angle of approach to Interaction design than we do, but our concepts of true mobility are identical.

We have opted to use the term "truly mobile" as it is logical, fits the intended use pattern well and has been used by other researchers. But mobility is a com-

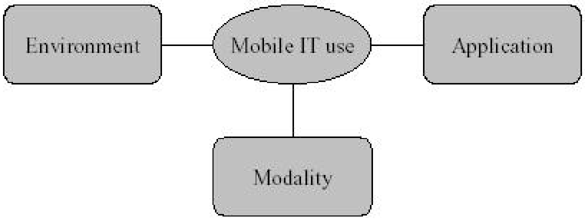

Figure 4.1: K&L Model overview

plex subject and has implications for all aspects of design. In the next section, we describe a prevalent theoretical framework of mobility and how this thesis fits into it.

4.2 Describing mobility by modalities

One model to describe mobility is that of Kristoffersen and Ljungberg (Kristoffersen & Ljungberg 1998*a*). In this model the modalities represents the fundamental patterns of motion whereas the environment is the physical and social surroundings. K&L argues that one of the fundamental differences between mobile and stationary IT use is that the stationary is reliable and static whereas the mobile use is unpredictable and heavily constrained in different ways. The mobile use today is largely based on the idea that a computer aided system shall offer the same benefits of that of an office - hence the “desktop” metaphor was adopted and taken into use by Xerox in the 1980s. K&L do not think that a handheld PC or a mobile phone offer the same basic functionality as a PC in the office, and therefore the desktop metaphor is inadequate. They (K&L) focus on effect rather than functionality. Effect is here described as what the user actually accomplishes. A research project named EMBASSI (Kirste et al. 2001) are working with a similar task. They argue that a paradigm shift is currently taking place; a transition from a function-oriented interaction with devices to a goal-oriented interaction with systems. Traditional systems today are often based on the user to chose functions to reach a goal. Different devices have different functions, similar functions in different devices behave differently. The EMBASSI article further shows agreement with K&L in that they say that it is not the functions that is important to select in order to get the desired effect, but the *effect* itself (goal oriented interaction).

The model above (figure 4.1) shows how K&L (Kristoffersen & Ljungberg 1998*a*) have organized and identified elements of mobile IT use. The three main components of mobile IT use are environment, modality and application. *Environment* is the physical and social surroundings. *Modality* is the fundamental patterns of motion. *Application* is the combination of technology, program and data you use.

An example of how the components work together can be described as this scenario; A new student at campus wants to send a SMS to a school friend nearby. Its an urgent message from the cashier-queue at a bookstore and the “caller” wants to know if he has found the correct book to buy for a lecture he is signed up for. The *environment* can be described as crowded and noisy (social), and the caller need to pay attention to the queue in progress. The physical surroundings, bookshelves, corridors makes it hard to give room to work with the mobile phone. Another dilemma is that the caller is carrying many books so he has just but one hand to operate the mobile phone. The *modality*, the fundamental pattern of motion, in this scenario is based on the fact that the caller is standing/moving slowly towards the cashier (not stationary).

Specifying a general criteria for an activity to be mobile or stationary is difficult because virtually all activities involves mobility of some kind. K&L suggest 3 modalities to describe the archetypes of mobility:

Wandering

Wandering is characterized by extensive local mobility. Local mobility is understood to be personal mobility within a certain area. One example is the IT support staff wandering around the building to help users with IT difficulties and also to receive new work tasks. Another example is tourists exploring an area of a city using a electronic map. We consider our experiment and prototype system to be of this mobility type.

Travelling

Travelling is an activity that takes place while travelling in a vehicle. This can be in a car, train, airplane and so forth. In this activity local mobility still can take place.

Visiting

Visiting is an activity that happens in one place for a coherent but temporal period of time. An example can be to visit a client to look over a contract.

One aspect that is by some considered a lack for the K&L model is the dilemma that the K&L model does not account for *how* a user operates the device (Pascoe et al. 2000). Beck et.al (Beck et al. 2002) takes this argument further and points out that the K&L model does not consider:

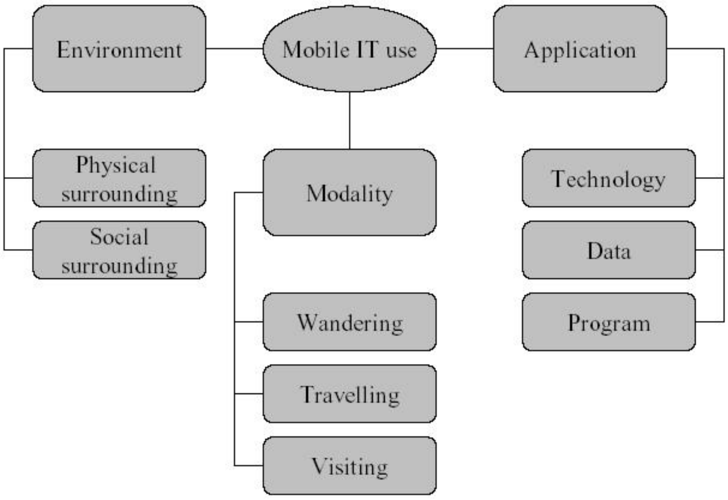

Figure 4.2: K&L basic reference Model of Mobile Informatics

1. What significance the user's motion has to the use-aspect; That is the relation between the activities and the mobile aspect of the user.

2. What influence do the surroundings impose on the user's activities; This refers to the relation between the activities and the environment surrounding the user and the mobile device.

3. What influence the state changes of the device has on the tasks carried out by a user; The correlation between the user's activities and the mobile aspect of the device.

Pascoe et. al. (Pascoe et al. 2000) also say the K&L modalities focus on the mobility of the user (wandering, traveling, visiting) and device (mobile, portable, desktop) and not the mobility of the activity. "*We believe it is the amount of mobility that the user requires whilst simultaneously using a device that is the primary factor in influencing its design.*" Although K&L's model is much used as reference, it is perhaps to general to provide meaningful data to a more specific research project. The relation between the modality, the application, and the environment may need to be addressed differently in each situation. In our case we found the model to be too abstract other than providing a conceptual framework, as was also the intention behind this model. The relation, as stated, may be found when one consider the activity of the mobility involved in a specific context.

4.3 Mobility in telecommunication

Consider this definition of mobility in contrast to the ones offered above:
"A radiocommunication service between mobile and land stations, or between mobile stations." (Institute of Telecommunications Sciences 1996).

This is how the Institute of Telecommunications Sciences, ITS, defines mobility. So far, we have discussed mobility in terms of how the end user experiences it. The Telecommunications Information Networking Architecture Consortium, TINA-C (TINA organization 2002), offers a highly technological model of mobility. TINA-C defines four distinct types of mobility that are applicable to telecommunications systems: terminal, personal, application and session mobility. This is how Thanh (1997) describes these four forms:

Terminal mobility

Terminal mobility focuses on the end-user's device for accessing the system. A system complying to terminal mobility allows the terminal to change location while maintaining all services.

Personal mobility

Personal mobility allows a human being to access or to be accessed by the network independently of where the access point and terminal used are located in the network and maintaining all services specified in the personal subscription.

Application mobility

Application mobility allows a software process to be relocated from one machine to another or even moved between machines while processing.

Session mobility

Session mobility assures that active sessions are not disrupted while terminals, persons or applications are moving or being relocated. However, sessions may be brought to a well-defined halt to be resumed later. Session mobility differs from application mobility in that the process may be transferred to a different type of device, for instance going from a desktop computer to a mobile phone.

For a mobile 3D system, the first two mobility types seems the most appropriate, as the intuitive way to implement such a system is on a mobile computer. But, as we shall see in the next section, there are alternatives to this approach.

4.4 Mobile computing vs. Ubiquitous computing

The inherent goal of mobility is to provide access to computer services irrespective of location. There are two obvious ways to accomplish this: either to have the user carry the computer with him at all times, OR to have computers available at every conceivable location. These approaches are representative of two competing models of the future of human-computer interaction: wearable computing and ubiquitous computing.

Ubiquitous computing

Proponents of the ubiquitous computing model see the world littered with "intelligent" objects, where computers are built into walls, tabletops and all manner of everyday objects. A person may interact with hundreds of computers at a time, each invisibly embedded in the environment and wirelessly communicating with each other (Falk & Björk 2001). The technological challenges that must be overcome in order to reach this stage are many and various, ensuring that the most extreme versions of ubiquitous computing are firmly in the realm of fiction. More limited versions of this idea are much more plausible, even attainable, using today's technology. Considering the amount of computers available to the public in densely populated areas, carrying a computer around with you seems something of a waste. For instance, at our campus there are publicly accessible computers in every building! There are still serious problems with implementing ubiquitous applications, even limited ones confined to, for instance, a university campus. Problems ranging from security and safeguarding personal information to technical difficulties in dealing with myriad personalized processes at the same time make this model impractical, at least for the time being.

Wearable computing

In contrast to ubiquitous computing, the wearable computing model endorses the idea that carrying a computer around with you everywhere you go will become second nature when computers are light enough, and interfaces sufficiently advanced, that you won't notice the computer at all. The notion is that of a highly personal device, always powered on and always accessible, serving a sole user throughout all aspects of daily life (Falk & Björk 2001). Needless to say, current technology has a while to go before this stage is reached. But already we see the emergence of wearable computing in such mobile devices as cell phones and PDA's. One of the greatest challenges to wearable computing is represented by the interface. Using these common mobile devices while performing other tasks still pose problems and even risk. For instance, today it is illegal to use a mobile phone while driving in many countries.

Both these models have their distinct advantages and drawbacks. Using TINA-

C's concepts of mobility, wearable computing emphasise terminal and personal mobility, and ubiquitous computing exemplify session and application mobility. The most likely scenario seems to be a hybrid of the two extremes, as described by Falk and Bjork (Falk & Björk 2001) and Rhodes et. al. (Rhodes et al. 1999). A first attempt of a ready-made synergy of the two can be found in the concept of content-based scalability, as implemented by the MPEG-4 standard (Moving Pictures Experts Group Visited 2003). Using this technique, multimedia presentations or applications alter their appearance and demands on the terminal and/or user to best suit the device they are viewed on. This happens dynamically and allows the user to switch to the most appropriate device as the situation and user demand dictates.

4.5 Location detection

As stated earlier, an important feature of Mobile IT is the ability to provide *contextual information*. Contextual information updates according to the user's surroundings, providing the user with useful information according to his situation. An example of a contextual information service is the weather service by DJuice (DJuice 2003) that gives users access to the weather forecast for their area. In order to provide contextual information, the system must have access to information about where the user is at a given time. Acquiring location data is often referred to as *location detection* and there are several such types of technology available. However, these existing technologies suffer from a variety of drawbacks: they are inaccurate or unreliable, or require bulky and expensive hardware.

The problem of accurate tracking is widely recognized. Accurate tracking has been the subject of a great deal of research, but position and orientation tracking is "one of the areas that has seen insufficient innovation in the past decade" (Zyda & Sheehan 1997). While it is true that there is no cheap and accurate location detection technology available for handheld computers today; there are existing alternatives, each with their own sets of strengths and weaknesses.

Another form of location detection is *orientation detection*. Sometimes it is desirable for the application to be aware of not only the user's location, but also his orientation. However, orientation data is both less essential in many cases and also more easily obtained with the aid of a compass. In this thesis, orientation detection is left entirely to the user of the system, and will be discussed in chapter xx.

Current technologies

A variety of ways to provide location data have been suggested: including InfraRed-dependent (IR) technologies and even ultrasound receivers (Harter et al. 1999). But the two most widely used techniques today are Global Positioning System (GPS) and mobile network cell identification. GPS and similar systems, rely on the mobile device receiving timed signals from a set of satellites. By measuring the relative delay between signals, the device can tell its distance from different satellites. As long as the signals from at least three satellites are received, the user will know his location with an accuracy of about five meters with current systems. Despite this relatively fine-grained resolution, satellite-based systems do have some drawbacks. Most importantly, GPS-like systems do not work indoors or in dense urban areas, as buildings block the satellites' signals (Bahl & Padmanabhan 2000). GPS-capable handsets are also quite bulky and expensive compared to mobile phones and PDA's.

In contrast, cell identification technologies take advantage of already existing technology in cell phone networks. Cell phone networks are built up around base stations that relay the cell phones signals. These base stations have a variable, but usually limited range. Cell phones within this area are considered connected to the base station. If a cell phone is connected to a certain base station it follows that the cell phone must be in the area covered by the base station. In densely populated areas, base stations have a very limited range due to the high density of mobile phones, down to 50 meters in some cases. In less populated areas, base stations can be 30 kilometres apart (figures taken from (Lähteenmäki et al. n.d.)). Cell identification is already in use in some networks like GPRS. Variants of the cell identification technique can be used with short-range networks like WLAN or Bluetooth. Using such a short-range network might still allow the user access to a global network, but there would be no guarantee of a smooth transition between base stations. The main drawback to cell identification is the limited and variable resolution available. Variants of the cell identification technology use triangulation or timed arrival schemes to provide greater accuracy. For a more in-depth comparison of the different location detection technologies see (ibid.).

In this section we have introduced location detection and the technologies available today. Location detection is an important prerequisite to providing contextual information, but there is no single definitive location detection technology available today. In the Augmented Reality chapter we describe how the location detection problem is connected to the type of contextual information called *annotations*. Location detection technologies are central to Andreas Butz et. al. (Butz et al. 2001) theory of the correlation between the resolution of location data and the graphical detail of navigation systems. This correlation and its implications for M3D systems are discussed in the Design Rationale chapter. Based on this correlation and the concept of annotations we present our own implementation of a location detection feature that relies on the user's recognition of 3D objects in the System Description chapter.

4.6 Mobile IT challenges

Mobile IT use introduces many challenges. Especially, the context of use, is central because of its dynamic characteristics. Grimstad et. al. as quoted by Maiken Solberg (Solberg 2001) describes 3 major complications for mobile IT use:

- **Complex environment:** The user may be outside and perhaps on the move. this situation differs significantly from sitting by a desk at the office. Unexpected interference makes less attention available for the wireless service.
- **Perceived pressure:** The user might feel he is under pressure and therefore be stressed and lose attention.
- **Less support:** It is not likely that the user will be able to take notes in a mobile situation, like he would if he was sitting by a desk in his office.

In an office environment the user do not always have to overcome limitations like short-term memory and attention span. More time and attention can be given a work task without compromising the risk of forgetting or down prioritize other tasks. In a mobile context these “other” tasks may be more important than operating the system itself - like paying attention to the car in front of you. The above list serves as arguments for designing interfaces on mobile terminals that minimizes the users’ memory load. There are of course many other complications with mobile IT use such as unreliable network, the technology constraints of the device or the cramped interface. We will take a look at some of these other complications in separate sections later in this chapter.

4.7 Handheld Multimedia technologies

In the previous sections we have presented some theoretical basics of mobility, in particular how it applies to handheld terminals. However, since part of our thesis revolves around design of a prototype system, we also need to look at the specifics of these handheld terminals. RT3D is complex and resource-intensive, so we must carefully consider how handheld systems differ from their stationary counterparts. This section deals with the specifics of handheld terminals that are of particular interest to developers of RT3D systems.

4.7.1 Mobile technical constraints

The technical limitations of handheld computers pose some serious challenges for developers of RT3D systems. This state is worsened by the fact that widely available handheld systems are a relatively recent phenomenon, with little standardization of hardware and software formats. These difficulties are not insurmountable, as there are techniques to compensate for hardware limitations and lack of established standards. The most important challenges are described below.

Low protocol stack

In the following discussion of technologies it is helpful to place them relative to each other within the terminal's software structure. This structure can be thought of as a protocol stack, since it shows the relationships between system components as a vertical hierarchy (figure 4.3). The system hardware is located at the bottom of the stack with several successive layers of software above it. A stationary computer has a much more complex protocol stack, with many more layers. This makes it easier to develop software that conforms to a particular layer, since it has more supporting software already installed and better-defined ways to interact with the other layers. The operative systems available for handheld terminals, on the other hand, work very closely with the central processor, with few software layers in between. This means that developers must make separate versions of their systems not only for each type of operative system, but for each operative system/processor combination. Since there are many different OS's and processors in use on handhelds today, this is a significant problem. Probably, the state of confusion with many available operative systems will not last for long. Already several smaller competitors, like Psion, have disappeared from the PDA market (IT-avisen 2002). The company BeOS Inc. merged with Palm Inc. in summer 2001. This may indicate a stabilisation of the mobile OS market. A more stable market will make the developer's job easier by allowing them to concentrate on the major OS's. There are platform-independent alternatives, the most common being Java. Most major handheld OS's offer or have planned Java support. Software written in Java can be run on any system that offers a Java Runtime Environment. The main drawback to this approach is speed. Java programs (in bytecode) run on top of a "Java Virtual Machine" (JVM), that in effect creates another layer in the protocol stack. This

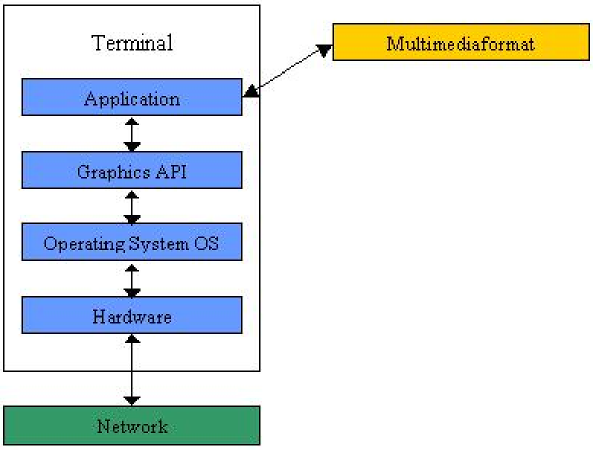

Figure 4.3: Schematic representation of the protocol stack of a handheld computer. (Illustration by the authors)

layer mediates between the Java program and the lower layer protocols and hardware and results in slower speeds. This is particularly felt on handheld terminals where processor resources are very limited.

No hardware acceleration

Many RT3D systems today require hardware graphics support to function properly, since such support is common in stationary systems. The most common example of this is modern 3D computer games as mentioned in chapter 3. Though it is possible to let the CPU do all the work on a stationary system through a software-rendrer this means a serious drop in graphics quality and speed. Unfortunately, graphics acceleration for handheld terminals will probably not become commonplace in the near future. This leaves software rendering the only available option for handheld terminals today.

Small screen sizes

An obvious limitation of handheld terminals is their small, usually liquid crystal, screens and low resolution. This limits the complexity of the material that can be presented. Mobile displays that compensate for this, like goggle-displays, are still some years away. However, software standards like MPEG-4 (Moving Pictures Experts Group Visited 2003) offer the possibility of displaying complex content on small screens of limited resolution. But it becomes very important to adapt the content to this medium. By designing the 3D objects to have high contrast in colour and shape the small display becomes easier to read. Scalable content will allow the content to be displayed in a manner best suited to the terminal. Most often this will mean that certain less-crucial elements will be dropped when the content is displayed on a mobile terminal.

Cramped interface

The interfaces of handheld terminals usually contain one or more of the following: a numerical pad, a touch-sensitive screen (usually coupled with a stylus for precision input), specialized buttons and possibly an alphanumeric keypad. The stylus/touch screen interface most common on PDA's today is best suited to 2D applications, as interface elements have a fixed size. In a 3D environment, elements grow and shrink according to perspective and angle. Also, this method of input means that the stylus and hand that holds it obscures a significant part of the screen every time it is used. 3D applications often require continuous input to "steer" inside the 3D world, which worsens the problem. Obviously, a more subtle form of input would be highly desirable. Some mobile phones already come with built in micro-"joysticks" (e.g. (Sony-Ericsson 2003)), partially to aid in game play. An interesting solution is offered by the 3D game Racing Days (Kitt Peak Inc. 2001) on PocketPC's. In this game, the user turns the PDA sideways, using his left thumb on the volume control button to accelerate/decelerate and using the stylus to steer by pressing a bar displayed along the bottom of the touch screen. This simple solution maximizes the available controls and minimizes the clutter caused by the stylus operation.

Slow processors

The processors available today for handheld computers have, at the time of writing, an upper limit of around 500MHz. For mobile phones the speeds are much slower. RT3D graphics are by their nature very processor intensive, especially with no graphics cards available. However, even though PDA's are slow in comparison to current stationary systems, in pure processing power they are equal to desktop PC's from a few years ago. And the gap is closing. There are indications that handheld computers will in the near future equal their stationary counterparts in all respects but size (Beck et al. 2002). This rapid development will probably extend to the other hardware technologies already mentioned, making tomorrow's systems more compatible with RT3D.

Slow and unreliable networks

The wireless telecommunications networks available today are relatively slow considereing the possibility to support real time transfer of complex 3D-data. Even the introduction of 3rd generation networks (3G) will probably not be sufficient to completely solve the problem. See Appendix B for a comparison of real and expected speeds of current and future networks. This appendix is especially useful since it uses our example system as the basis for its calculations of speeds. Note that this applies to long-range telecommunications networks. For short range wireless technologies, the situation is different. Existing short range networks such as Bluetooth and WLAN offer higher transfer rate and are readily available. Our example system simulates the use of such a short range wireless network. The limited range of these networks had a profound impact on our system's capabilities and had to be considered during its design. See chapter 8 for a detailed description on how we propose to integrate a short range network and how this affected our system.

Wireless networks are often said to be unreliable. Transmission over the air involves many uncertainty factors like the weather condition, transmission interference, and obscure signal.
In a mobile use setting, the mobile device needs to reconnect to new networks as it passes in and out of network "zones". The shift from one network to another may take a little time. And one cannot know if the quality or bandwidth of the new network will support the service running on the mobile device. For instance: switching from a WLAN network to a GSM network while attending a video-conference will reduce the quality (at best) or crash the service (at worse).

No available graphics API's

An API (Application Program Interface - and sometimes spelled Application Programming Interface) is the specific method prescribed by a computer operating system or by an application program by which a programmer writing an application program can make requests of the operating system or another application. A graphics API is a software "toolbox" that provides already defined ways to describe and manipulate graphics data. 3D graphics API's are routinely used in the development of commercial 3D systems. Essentially a 3D API is a piece of software that significantly improves and eases development of 3D-systems. Unfortunately there are currently no available graphics-API's for handheld terminals, which slows development of new 3D programs. At the time of writing, however, a consortium of software developers announced the creation of GAPI, a 3D graphics API for Pocket PC's that are especially targeted towards games manufacturers.

No dominant multimedia standard

A multimedia standard is a predefined method for coding multimedia content. These standards are developed by large, international organizations like W3C (World Wide Web Consortium 2003) and MPEG (MPEG Commitee website 2001) and aim to ease interoperability between applications. Examples of such standards include the MPEG standards (ibid.) and SMIL (SMIL - Synchronized Multimedia Integration Language 2001). These standards typically have different emphasis.

4.8 VRML

To ensure interoperability between programs, several organizations have standarized unique ways of communicating with- and transfering of multimedia content. Some of the leading multimedia standards today include SMIL (SMIL - Synchronized Multimedia Integration Language 2001), X3D (X3D - eXtensible3D 2001), MPEG-4 (Moving Pictures Experts Group Visited 2003) and VRML (VRML2.0 spesification 1996) which will be described in this chapter. These standards provide functionality such as syncronization, scene-structurization and compressing for multimedia content.

VRML is well documented and widely used on Internet. There are many different VRML browsers to choose between and most importantly there is a mobile VRML browser application called Pocket Cortona that is able to interpret and render fully VRML worlds on a Pocket PC (PPC). During our work and pre-study for the firm Octaga AS we were introduced to VRML and mobile 3D. With this background we found VRML adequate for our purpose. In addition VRML is not a proprietary standard meaning, it is free to use and the entire standard is open for study.

VRML is an acronym for Virtual Reality Modeling Language which is used to

present 3D graphics on the web. It is similar to HTML in that it is simply a text file, but the contents of a VRML file are definitions of objects and their attributes that make up a world.
In addition to the 3D objects, VRML can be used to create animation sequences by using it's scripting language. Using the scripting language, a programmer can write a procedure that will change the position of the camera over a certain interval of time.

In this chapter some basics of VRML and its scripting possibilities will be described. The complete VRML- and script code for the testbed system can be read in Appendix A.

4.8.1 Brief history of VRML

The first VRML version, VRML 1.0, was based on a Silicon Graphics ascii file format called "Open Inventor"(Silicon Graphics 2002). Rikk Carey and Gavin Bell adapted the Inventor File Format for VRML, with extensive input from the rapidly growing www-vrml mailing list, and the support of Silicon Graphics, Inc. This work began in 1994 and led to the version most used today, VRML 2.0, in 1996. The specification was then formaly accepted by the ISO organisation in 1997 with short name VRML 97 and formal designation ISO/IEC 14772-1:1997 (VRML2.0 spesification 1996).

The initial requirements for VRML, which is still valid, are:

- Platform independence
- Extensibility - the ability to add new elements.
- The ability to work over low-bandwidth (14.4 kBps modem) connections.

More requirements were added during the spesification period:
Authorability
Make it possible to develop application generators and editors, as well as to import data from other industrial formats.

Completeness
Provide all information necessary for implementation and address a complete feature set for wide industry acceptance.

Composability
The ability to use elements of VRML in combination and thus allow re-usability.

Implementability
Capable of implementation on a wide range of systems.

Multi-user potential
Should not preclude the implementation of multi-user environments.

Orthogonality
The elements of VRML should be independent of each other, or any dependencies should be structured and well defined.

Performance
The elements should be designed with the emphasis on interactive performance on a variety of computing platforms.

Scalability
The elements of VRML should be designed for infinitely large compositions.

Standard practice
Only those elements that reflect existing practice, that are necessary to support existing practice, or that are necessary to support proposed standards should be standardized.

Well-structured
An element should have a well-defined interface and a simply stated unconditional purpose. Multipurpose elements and side effects should be avoided.

Today the VRML standard can deal with a wide range of multimedia content. Text, sound, images, and even video can be used together to make out a VRML scene - or a VRML world. [1] VRML is often used for modeling and simulating purposes but it is also taken into use by virtual communities offering a virtual world as frame for communication - e.g. (Cyber Town 2002).

4.8.2 Building a VRML world

In VRML the main building blocks are called nodes. These are structured hierarchically to make up the world. Objects can be grouped into more complex objects, used multiple times, translated, and rotated. Since there are several good tutorials and articles on how VRML works, (i.e. (Overview of the VRML2.0 manual 1996)) we will only present some basic consepts in this section.
VRML (*.wrl) files have 3 basic elements:

1. A header which tells the browser that the file is VRML and which version is

---

[1] The distinction between a scene and a world is obscure. We like to think about a scene as a presentation, and a world as a 3D virtual “landscape”. Allthough the word “scene” or “scene-graph” is used in standards like VRML and MPEG-4 we use the word “VRML world” throughout this paper to emphasize the “world” characteristic.

used. A header line is mandatory.
Example:

```
#VRML v2.0 utf8 Generated by Terje’s v0.2 generator
```

“utf8” is a unicode that defines a certain character set. After the unicode, there is space until end of line for comments.

2. Comments are preceded by a # and last until end of line (with the exception of the first mandatory line).
3. Nodes describes objects and their properties and makes out the majority of the .wrl file. The nodes are organized hierarchically in a scenegraph to provide an audio-visual representation that participate in the event generation and routing mechanism.

A simple VRML example is that of a box.

```
#VRML V2.0 utf8 =- Executable simple texture box -=
DEF box Shape {
   appearance Appearance {
      texture ImageTexture {
         url "adminbygg.jpg"
      }
   }
   geometry Box { }
}
```

The example shows a node named “box” that is defined through two fields; “appearance” and “geometry”. The appearance field contains an Appearance node which loads an image-file, adminbygg.jpg, and wraps the texture around the object. The geometry field contains a Box node that spesify the geometry as box-shaped. The hierarchical structure is easy noticeble in this example. hierarchicly under the Shape node is the Appearance node and the Box node. The Appearance node has one additional underlaying node, the ImageTexture node. Together we say that these nodes describes the box *object*..

A VRML world contains nodes which describe objects and their properties. The world may have zero or more root-nodes. The coordinate system in which the root nodes are displayed is called the world coordinate system. In our model of Blindern campus all buildings and other objects is accurately placed within this coordinate system to present the actual orientation.

Conceptually speaking, every VRML world contains a viewpoint from which the world is currently being viewed. Navigation is the action taken by the user to change the position and/or orientation of this viewpoint thereby changing the user’s view. This allows the user to move through a world or examine an object. Viewpoints can be defined manually by using a node called Viewpoint. More on navigation and how we designed viewpoints in our world can be read in Appendix A as well as in the system description section 8.5.

4.8.3 Scripting

For advanced interaction or animation, a VRML world can change dynamically by the use of scripts. Allthough this feature (ECMA code) is not described in detail in this paper we made heavy use of the ECMA script language as can be seen in the Appendix A's definition files. Chapter 8 gives insight in how we designed our system.

4.9 Summary

We have in this chapter described mobile IT use from different angles. The context of use, categorized by Kristoffersen and Ljungberg by modalities and mobility seen from the telecom industry both address problems of the dynamic use-context. Wheras K&L looks at mobility from the end-user point of view, the telecom industry have a rather technological approach. Fields of research address mobile complications in different ways. We have described ubiquitous computing and wearable computing as two distinctively different fields that both seeks to add advantages and overcome mobile difficulties. Our project has the most in common with the latter viewpoint. One key aspect of mobility is the contextual dimension it offers. Location detection may capture aspects of this context by enabling the mobile device to detect the surroundings of the user. The range of new possibilities in mobile computing carries along challenges. The shared attention for the user as well as the technological constraints of the mobile device and networks are not trivial. A developer needs to know of the difficulties and design with them in mind.

Standardized ways of presenting content for mobile services are beginning to emerge. One such standard for presenting multimedia content is VRML. Mobile VRML browsers are currently available offering cross platform independent possibilities.

Chapter 5

Interaction Design

FOR most people, everyday life entails operating quite a number of electronic devices like VCR's, TV's, mobile phones, ticket machines, etc. Not all interfaces are easy to interact with. Sometimes a manual or first hand experience with the device may enhance the usability, but sometimes the interaction is just too poorly designed to be of any use for the average person.
Interaction design is defined by Preece (Preece et al. 2002) as:

> *.. designing interactive products to support people in their everyday and working lives.*

In order to design for interaction, usability testing and evaluation are of great importance, and it is recommended that they occur during all stages of the product's development (Preece et al. 2002, page 339). Due to the nature of our experiment where 3D and shared attention are set in focus, we see usability testing as supplementary technique rather than the focus of our experiment. Our aim is not to test the "product" (as is the term used in the definition of interaction design above) as a whole but rather the 3D part of the system and its effects on shared attention. As stated in our introduction chapter, we believe that proper design is crucial to make a mobile 3D system usable. For this reason our focus in this chapter will lie on interaction design rather than usability testing.

Some believe that there are certain universal guidelines or rules of thumb that are valid for practically all computer systems. These guidelines may provide advice on the solutions of design problems. Some of these guidelines are very general and addresses problems that should be dealt with by a system developer. For instance making a UI intuitive is one rule of thumb that challenges the developer in many ways. To deal with such a rule poses many considerations. Not only does the developer need to take into account what usergroup the system eventually is to be used by, but the context of its use and also ethical issues among others must be adressed. Often guidelines apply to a certain context and one needs to modify them and adopt in order to make use of them. Pioneers in the field of user-centred design are Jacob Nielsen, who presented a set of interface design guidelines, or heuristics in 1993 (Koenen 1993), Ben Shneiderman who “presented his eight golden rules” in 1992 (Shneiderman 1997) (Almstrum 2003) and the “visual seeking mantra” (Shneiderman 1996) and Donald A. Norman who explores the design of everyday things (Norman 1988). However studying these rules we found them too general to be of much use in designing our very specific mobile system (as is also commented by others, e.g. (Borälva et al. n.d., page 7)). Instead we draw on research projects that are more in-line with ours.

According to Pascoe et. al. (Pascoe et al. 2000), two essential features of mobile systems that warrant special attention during interaction design are shared attention and context sensitivity. Shared attention can be viewed as a "downside" to dynamic context: taking your computer with you anywhere means that there are other things to consider beside the computer. Both the utilization of dynamic context and minimizing shared attention are technical challenges, at least to a degree. Utilizing dynamic context is more applicable to system design and so is discussed in the appropriate chapter. While designing for shared attention is an issue that applies to interaction design.

This chapter present our theoretical basis of mobile interaction design. We start by presenting the consept of metaphors and mental models. Then, input-/output techniques and dynamic user interfaces are described light on mobile informatics. Finnaly we draw on guidelines from other related research.

5.1 Metaphors and mental models

One of the difficulties in designing a user interface is described by Brad A. Myers as “the inherent complexity of tasks and applications” (Myers 93). Complexity is here understood as the difficulties in presenting the user with the possible functions of a system. Some systems, e.g. AutoCAD, can have more than 1000 functions. In mobile IT use, high complexity of a system influences the workload and increases the system’s demand for attention. One should therefore seek to *minimize the complexity.*

One way to overcome complexity, Myers say, is to use metaphors that exploits the users’ prior knowledge by making interface objects appear like objects that the user is familiar with. Metaphors is a way of describing conceptual models (Preece et al. 2002, page 55-60). By this is meant a conceptual model that has been developed to be similar in some way to aspects of a physical entity but that also has its own behaviours and properties. A GUI-component can describe a conceptual model through a metaphor. In a similar fashion, the GUI components of an entire system can be tailored to create a higher-level conceptual model. This is often re-

ferred to as a *mental model* (e.g. (Marcus 2001), (Preece et al. 2002, page 92-95)). Metaphors in the form of GUI components may together help express this mental model. An example to help distinguish between metaphors and mental models is the mental model of a desktop widely used in many operating systems today. This mental model is expressed by using metaphors such as folders, files, and a trashcan. Although the desktop model is commonly referred to as "the desktop metaphor" it is more accurately referred to in this context as a mental model.
In the following section we describe metaphors and mental models in more detail.

Metaphors

In computer systems metaphors are used to conceptualize abstract, hard to imagine, and difficult to articulate computer-based concepts and interactions in more concrete and familiar terms and as graphical visualizations of the interface. However the inevitable mismatches of the metaphor and its target are a source of new complexities for the users (Carroll et al. 1988). A mistake sometimes made by designers is to try to design an interface metaphor to look and behave literally like the physical entity it is being compared with (Preece et al. 2002, page 55-60). According to Preece et. al. this misses the point about the benefit of developing interface metaphors, because they are meant to be used to map familiar to unfamiliar knowledge, enabling users to understand and learn about the new domain. Other criticisms and difficulties that adds to the complexity of using metaphors include the following: (reformatted from (Preece et al. 2002, page 55-60))

- metaphors are often culturally or logically misinterpreted
- metaphors may be too constraining
- metaphors may conflict with design principles
- metaphors may be poorly chosen or designed so that the user may not understand the underlying systems functionality beyond the metaphor
- Metaphors may limit the designer's imagination in conjuring up new paradigms and models - designers may fixate on overused ideas

Mobile systems have been critisized as being influenced by the mental model of a desktop (Kristoffersen & Ljungberg 1998*b*). This mental model was initially meant to provide a user with the familiarity of an office setting. Perhaps metaphors in mobile systems need to be refined?

In a paper that deals with mobile maps (Rakkolainen et al. 2000), the researchers found that the metaphor of a book provided the necessary conceptual model for their user interface. Because of the small screen size of the mobile device they chose to use interleafs and palettes for easy switching between the modules the system provided. They emphasise that such a metaphor will provide a better solution compared to a menu-driven interchange between the modules.

In 3D worlds the use of metaphors is widely adopted. In the usability study of a 3D world (Köykkä et al. 1999), conducted by using Nielsen's heuristics, one of the outcomes was the proposal of new heuristic categories. Especially metaphors were considered important in that they *"... have to be clearly understandable"*.
In our experiment the majority of the subjects did not immediately accept the compass metaphor. In the discussion chapter we discuss several possibilities of why this happened, and how this can be understood from a technical point of view.

Mental model

The human-computer interface mediates between the user and the computer system. It protects the user from the "harsh realities of the system", reflects the system model to them and translates their intentions into appropriate system activity. The user forms a model, known as the user's mental model, of how the application works. This model forms the basis of future interactions with the system and enables users to predict system performance (Faulkner 1998, page 54). Metaphors are important elements in translating a mental model to the user. By i.e. using well-known metaphors (for a particular user group), a mental model can be formed helping a user to understand the system.

With respects to our prototype system a natural metaphor would be a map/guidebook. The mental model of a map or a guidebook is in accordance with the points from Preece (Preece et al. 2002). I. e. that a metaphor should map familiar to unfamiliar knowledge and that one should not try to design an metaphor to behave literally like the physical entity it is being compared with. Though Preece's points are intended for metaphors, they may be applied to mental models as well.

In a research paper that addresses tourists use of tools such as maps, guidebooks and oral inquiries, (Brown & Perry 2001) Brown et. al. observed that: "Whereas a map shows the physical relation in space a guidebook brings together place and space by combining physical and social information." In our prototype system we present both map data (3D model) and additional information (through photorealism and task related information) in conjunction. A guidebook might, in our case, provide an appropriate mental model. Another interesting finding from this research paper was that the tourists often needed some sort of link between the different aids to fully take advantage of the information:

> *... it was only in combination with local details [through oral inquiries] that they [the tourists] could make these [maps and guidebooks] work to help them plan and enjoy their visit. The tourists needed a mapping between the map and the physical world.*

This encouraged us to believe that 3D graphics in combination with added information (e.g. annotations) might provide just such a link. Using a guidebook as mental model makes the user expect that the system will provide physical as well as social information and also show the link between them. This was also in keeping with our approach inspired by Augmented Reality (see section 3.2). We believe that photorealistic 3D graphics with added annotations might provide both social and physical information. Further, the 3D space provides opportunities to address the link between them because annotations can be mapped directly to the physical objects. An interesting finding from our experiment, however, was that the subjects did think of our system as an “advanced/electronic map system” (we observed this during the thinking aloud method). This could be for a number of reasons that we will not debate further, due to the boundaries set by our problem area. As explained in chapter 9 we did not add more “additional information” than necessary for the subjects to complete their tasks.

Another meaning of the term “mental model” is more intuitive. A “mental model” is often used to refer to a construct committed to memory. A simple example of such a construct could be a rugby ball. It is relatively easy to remember the size, colour and other qualities of such an object, and imagine what it would look like from a certain angle. In this thesis we will refer to such a mental construct as a “mental map”, to distinguish it from “mental models” and to focus on its orientational aspects.

There were some indications that the subjects were able to build such a mental model, or rather a mental map, of the 3D world. Memorizing 3D objects relative to each other from the 3D world decreased the need for attention to the system.

5.2 Input/Output techniques

A very basic interaction design decision is how to handle system input and output. Usually the computer handles input/output through visual, audio or tactile channels. These channels are sometimes referred to as input/output *modalities*. In a truly mobile system, the choice of input/output modalities can have a great impact on the user’s attention. The dominant sense of human beings is sight. Consequently, a system that relies on visual output is interfering with the user’s primary means of experiencing the real world. Other truly mobile systems have explored this connection and have devised ways to utilize other modalities than sight in truly mobile systems.

5.2.1 Using other modalities than sight

In the article “Using While Moving: HCI Issues in Fieldwork Environments” (Pascoe et al. 2000), Pascoe et. al. formulates a general principle for truly mobile interface design: Minimal Attention User Interfaces (MAUI’s). MAUI’s are Pascoe et. al.’s expression for a technical solution to what we call the shared attention

Figure 5.1: Image from the article "Using While Moving" showing the fieldworker in his environment.

problem. According to Pascoe et. al. a MAUI is designed with two characteristics: dynamic user configuration and low attention capacity. The former is discussed in more detail in the next section. The latter characteristic of MAUI's; low attention capacity lies at the heart of the shared attention problem. Recognizing that the use context of the field worker implicitly entails a shared attention problem, the challenge for the Pascoe team became how to compensate for this in their interface design. Their finds can be summarised as one-handed interface, eyes-free interface, task layering, use of the stylus interface and task automation Several of these finds were both applicable and useful to us in designing our own system's interface. Most importantly task layering, use of the stylus interface and task automation. The first two, however, reflect Pascoe et. al.'s position that a truly mobile system based on visual output only would be too distracting. As a suggested solution, they describe a system that relies solely on audio output and tactile input.

Kray and Porzel's article "Spatial Cognition and Natural Language Interfaces in Mobile Personal Assistants" (Kray & Porzel 2000) is another example of using other modalities than sight as a solution to the shared attention problem. Here they design a system similar to the one described in this thesis, but focusing on hearing rather than sight, The "Talking Map" system they describe uses vocal input and audio output to receive the wanted destination as input from the user and giving a verbal description of the route there. The advantages to such a system are obvious: using verbal cues frees the user's most important sense, vision, to concentrate completely on his surroundings. The output and input devices (earphones and microphone) are perceived as being less intrusive than AR-goggles or a PDA screen (ibid. pp1).

However, this design looses the ability to link annotations in 3D space in the direct way that a visual system can. The “Talking Map” system includes functionality to provide verbal information about significant features (ibid. pp3). But care must be taken so that delivering unambiguous verbal directions to identify these does not become laborious and time consuming.

Also, though perceived as less intrusive, an audio interface holds the possibility of creating another shared attention problem by drowning out other audio signals. An audio interface does not completely solve the shared attention problem as mobile phone users have witnessed. Concentrating on stimuli from other modalities than sight can prove distracting as well.

In contrast to the systems described above, ours is designed around the concept of truly mobile RT3D. 3D is exclusively visual, so relying on other modalities than sight is not an option. It could be possible to design an M3D system that relied on other modalities in addition to sigh, but investigating this option was considered less significant to our problem area. Pascoe et. al. and other researchers point out that there are other methods that can help deal with the shared attention problem. Perhaps the most important of these is the dynamic user configuration.

5.2.2 Dynamic user interfaces

From Pascoe et. al’s definition of MAUI’s we have the two characteristics: dynamic user configuration and low attention capacity. Dynamic user configuration means that the user is able to optimize the interface to his current situation. For instance, the optimal method of interaction will probably be very different when the user is sitting at a table than when he is walking across the savannah observing giraffes. The type of data that is manipulated will probably be different as well. Pascoe et. al. designed two specific user interface modes, one for a stationary use context, giving the user full access to all data and functions, and a reduced interface that was optimized for an ambulatory use context giving the user limited access to data and functions. This approach worked very well for the subjects in Pascoe et. al.’s study: They found the system to be easy to learn and easily managed to switch between the two interface modes.

Other authors have also seen the advantages of a user interface that tailors itself to the user. Jameson et. al. point out in (Jameson et al. 1999) that such systems have broader appeal than just mobile systems. But such a system would have a definite impact in a mobile setting as the users resource limitations are constantly changing. As explained in chapter 6, as a user’s attention to the system drops, one can expect that the quality or magnitude of input decreases. Offering several different modes of interacting with the system based on the user’s available attention, like Pascoe et. al., could be a great benefit. As in the fieldworker example, a user that has little attention to spare can be offered a very simple user interface with only very basic functions. This would prevent the user from wasting time selecting between options and allow for faster input. Whereas to perform complicated tasks, the system can assume that the user has more attention directed at the device and so can offer a greater variety of input selections.

In our system, offering users a set selection of ways of interacting with the 3D model optimised for different use-patterns was deemed to be too limiting. Our goal was to investigate how the users reacted to this new tool, rather than force them into use patterns that we considered optimal. Since the focus of this thesis lies on mobile 3D, the most interesting area to provide different user configurations would be in manoeuvring in the 3D world. We therefore opted for providing several navigation modes within the same user interface. Each of these modes would give the user a different degree or method of controlling and viewing the 3D world. Each navigation mode can be seen as a specialised user interface configuration. The downside was the risk of confusing the users with too many options. In chapter 7 we show the options we selected to incorporate and the rationale behind them.

5.3 Other user interface lessons

The MAUI's of Pascoe et. al. are "... not limited to eyes-free forms of interaction but also covers other methods that attempt to minimize the amount of distraction caused by the user's activity" (Pascoe & Ryan 1999, 426). There are many interface design experiences that these and other authors have made that are directly applicable to our system. In this section we describe these.

Layering of tasks

An important guideline that Pascoe et. al. employed was that minimizing attention demand does not mean minimizing the amount of interaction. In fact, one proposed means of minimizing user attention is limiting the available input variables. Whereas a stationary system might present the user with over a dozen buttons to select from, a minimal attention design could group and layer these choices to make them easier to choose between and remember. The giraffe system used a layering technique that was based on only three choices available to the user at any given time. Selecting one opened for a further three choices and so on. This required less attention from the user in that the display was simpler and the actions involving selecting the desired result were easier. This did not necessarily make the system quicker to use, nor did it allow the user to interact more efficiently with it, but it allowed more of the user's cognitive resources to be applied to other tasks.

In our system, layering of selective tasks became an important technique in dealing with annotations and help functions. This approach was found to have an added advantage in an M3D setting, as screen space is even more at a premium than in other mobile systems. Layering selection tasks allowed menus to take up a small and fixed amount of space.

Stylus interface

One of the most common way to interact with a PDA is a stylus and touch-sensitive screen combination. Pascoe found the stylus interface very useful when compared to flip-open PDA/phone hybrids or miniature keyboards. One of the main advantages of the stylus is that it offers a very ergonomic interface solution while the subject is standing upright. It is also very flexible, allowing system developers to delineate areas on the screen that can be touched by the stylus to produce system responses. An additional advantage that Pascoe e. al. does not mention is its very direct means of manipulation. The touch screen/stylus interface allows the user to perform operations directly on the screen, rather than using an external input device. This is ideal in a shared attention situation. However, as explained earlier, the touch screen/stylus interface offers its own challenges to an M3D system.

Task Automation

Another way of minimizing attention demand that Pascoe et. al. pointed out was automating certain tasks or parts of these. According to Pascoe et. al. automation of tasks aids high-speed interaction and thereby helps dealing with shared attention. In the Giraffe study, entering certain types of information, such as the time of day was done automatically. Of special interest is the relationship they point out between automating tasks and context awareness. If the system is context aware, it can automate the task selection itself. For instance, if an ecological field-worker system detects that the user is located on a beach, it can automatically prompt the field worker to enter the position of the tide. Automating tasks or partially automating them became one of our most important tools in designing our system's "MAUI".

Additional principles

In a paper that explores heuristic guidelines for 3D multiuser worlds (Köykkä et al. 1999) the authors claim to have found three important principles a system developer should take into account when designing a 3D system. These “new” principles, the authors explains, should be added to the 10 heuristics defined by Nielsen to address and strengthen the usability of 3D-world systems.

1. Real world metaphors have to be clearly understandable

2. Provide support for orientation, navigation and movement

3. Avoidance of delays and waiting in periods in the performance

One can argue if the new principles above is taken account for in Nielsens heuristic principles or not, or if they can even be considered balanced along with the 10, but it is nevertheless clearly that a 3D world requires design rationale beyond the 10 traditional heuristics. For instance navigation means in traditional interaction (window interaction, stylesheets interaction, webpage interaction etc) involves mouse or use of arrow buttons to navigate which are perhaps thought as so simple to use that they are not accounted for in the 10 heuristic principles. Clearly, interaction mechanism includes more than virtual buttons and screen display, but also the artefact design place an important role, escpecially when attention and limited interaction means (e.g. one-hand operation) are important as in a mobile situation. In a 3D world, navigation is essential because the navigation is one key aspect of interacting with the system. We therefore found the above principle number two an important principle to take into consideration in designing for navigation in a 3D system. Although to provide a navigation support is not always enough: we feel that the user needs several navigation possibilities - e.g. drag the pen to navigate the world should be accompanied by the possibiliy to use the hardware buttons on the artefact as well. Using a pen typically involves using both hands to operate the device - one hand to hold the pen, and one hand to hold the device. Another argument is that navigation in 3D space often requires two input mechanisms, one to manipulate the perspective, and one for movement.

Guidelines for virtual environments were also considered. An article that lists guidelines that deals with metaphors and navigation in virtual environments is that of Vinson (Vinson 1999). These guidelines are specific to large-scale virtual environments, and assume that the 3D world created is not based on a real-world setting. Although we tried to apply Vinson’s design guidelines, these were not always applicable. For instance, one of his guidelines state that “Landmarks should be visible at all navigable scales”. This was simply not possible to achieve in the “Birdseye” viewpoint, as this viewpoint was intended to simulate the top-down perspective of a traditional map. The limited resolution of the iPAQ meant that at this scale, only the buildings were distinguishable.

This article did give us some incentive to maintain realism within our 3D world: When it comes to designing landmarks, concrete objects work better than abstract ones. When navigating 3D worlds, test subjects prefer to use 3D objects like model cars and forks to abstract art to orient themselves. “It was felt that the 3D objects were easier to remember than the abstract art and that this accounted for the difference in navigability” (Vinson 1999). This can be seen as an indication that realistic object representations are preferable to stylized ones because they are perceived as being more “unique” and hence easier to remember. This is, as mentioned above, provided that the real setting being modeled contains objects that are clearly distinct.

Faulkner (Faulkner 1998) presents a set of guidelines for developers of systems that rely on shared attention. These guidelines relate directly to the shared attention problem and so are examined in the following shared attention chapter, chapter 6.

Chapter 6

Shared attention

Everyone knows what attention is. It is the taking possession by the mind, in clear and vivid form, of one out of what seem several simultaneously possible objects or strains of thought. Focalization, concentration are of its essence. It implies a withdrawal from some things in order to deal with others.

– William James, 1890

ATTENTION has been applied to a wide range of phenomenon. Sometimes it is used to refer to the ability to select part of the incoming stimulation for further processing, but it has also been regarded as synonymous with concentration. It has also been applied to search processes in which a specified target is looked for and it has been suggested that attention is dependent on alertness (Eysenck 1984). However, attention is most commonly used to refer to selectivity of processing, and this is how the word is used in this thesis.

Shared attention stands at the heart of our problem area. Whether an M3D system is successful or not depends in a large part on the user being able to utilize the system while interacting with the real world. This chapter goes into some detail in trying to describe what is meant by sharing attention, as well as how it applies to M3D systems.

There is an important distinction between focused attention and divided or shared attention. When using focused attention, a person is concentrating on a single stimulus and ignoring all distractions. Examples from daily life could be studying for an exam or watching a movie. Whereas when applying shared attention, a person is processing two or more stimuli at once. Two forms of shared attention are commonly referred to: *dual-tasking* and *multi-tasking*. When dual-tasking, a person performs two tasks simultaneously. An example of dual-tasking could be driving a car and using a mobile phone (Strayer & Johnston 2001). However, when multitasking, a person carries out a number of tasks during the same period of time by alternating between them (Preece et al. 1994, page 105). Note that a person can be performing more than two mental tasks at once and still be considered dual-tasking. The distinction lies in whether the mind is indeed performing two processes at once (dual tasking) or switching between focused attention on a number of tasks. This definition comes from Beck et. al. (Beck et al. 2002) since this is the work we refer to when discussing the topic.

What is here referred to as shared attention is sometimes known by other names. In some publications, particularly those with a focus on psychology, the term *divided attention* is used. In this thesis, divided and shared attention are synonymous. We have chosen shared attention as the term we use, but when referring to the work of others, we will quote them on their preference.

In this chapter the reader is first introduced to the current cognitive models of human attention. This introduction is very brief and is primarily intended to define the terms used later as well as serving as a point for comparison. In the next section, we describe a framework for attention models that is used in the field of human-computer interaction. Narrowing down to the area of interest, the shared attention problem inherent in truly mobile IT is examined. In section 5, the information from previous sections is collated and applied to our specific problem area: shared attention in mobile 3D systems.

6.1 Shared attention in cognitive psychology

The branch of psychology that is concerned with cataloguing and quantifying human mental processes is known as *cognitive psychology*. It is this branch that deals most directly with human cognitive abilities, and therefore shared attention. We rely on the current models of cognitive psychology to establish some "ground rules" for shared attention.

According to Eysenck (Eysenck 1984) the original model of shared attention in cognitive psychology stated that the difficulty of performing two tasks at once was equal to the sum of their difficulties. More recently, it has been recognized that the interplay between simultaneous tasks can have an adverse effect on their execution. As the understanding of human cognitive processes has become more sophisticated, our models of the human mental resources have become more refined. It is now believed that simultaneously performed tasks can compete for specific mental resources and thus interfere with each other.

Based on this model of interference, cognitive psychology has identified many factors that affect dual-task performance. These form the foundation of many shared-attention theories in HCI and are summarized below. The factors listed here are from (Eysenck & Keane 1995, page 108) as quoted in (Beck et al. 2002, page 35).

Individual task difficulty

Regardless of the interplay of the tasks, it seems logical that if the tasks are difficult to perform separately they will still be difficult if performed together and vice versa. This is such a basic criterion that examples are unnecessary.

Practice and experience

One factor that has been determined to have a heavy influence on dual tasking is the person’s familiarity with the tasks. Both separately and when performed together. From this it follows that if the individual tasks are easy to learn separately, they become easier to master together.

Degree of similarity

Perhaps in spite of intuition, the more similar two tasks are, the more difficult they are to perform simultaneously. Consider for instance patting yourself on the head while rubbing your stomach. Performing these tasks appears simple when first contemplated, but because of their similarity are difficult to execute concurrently yet independently. Eysenck (Eysenck 1984) divides similarity into four groups according to the mental processes they stress. The first deals with the timing and resource allocation of the three stages of mental processing. The remaining three focus on the strain the tasks put on the individual process stages.

Similarity of process stages

All cognitive activities follow the pattern of registering input, perform internal processing and output. Input is registered by the senses, the stimuli are processed by the brain and then the brain's output takes the form of gestures, vocalizations or similar. If the simultaneous tasks are in the same process stages, that is: they have similar timing, then they compete for the same mental resources and can interfere with each other.

Similarity of stimuli

Whether visual, auditory or relying on other senses, attending to different stimuli of the same kind is more difficult than if they had relied on different senses. The dominant sense a stimuli registers on is sometimes called its *modality*. Tasks that rely on different modalities for input are easier to perform simultaneously than tasks that rely on different modalities.

Similarity of internal processing operations

Just as similarity in input modality can cause tasks to compete with each other, similarity in the demands they put on the cognitive processing itself can cause confusion and error. For instance, if both tasks are of a mathematical nature, they are more likely to become intertwined in one's mind than if one relied on, say, voice recognition.

Similarity of responses

The mirror image of the first factor, it is generally more difficult to perform output tasks that are similar than ones that use different modalities. This is exploited by many tests given to applicants for jobs that demand high level of competence during periods of stress. An applicant can for instance be given a red and blue pen and required to make a blue cross on a sheet of paper when he hears one sound and a red circle when he hears another.

Using mental processes as a guide to determine the similarity between tasks is hardly trivial when the tasks in question have little in common. How similar are piano playing and poetry writing, or driving a car and watching a football match? Only when there is a better understanding of the processes involved in performing such dissimilar tasks will sensible answers be forthcoming (Eysenck 1984).

The factors listed above are very general, and in applying them to interface design some modifications are necessary. Faulkner (Faulkner 1998) translates these criteria into a more manageable set of guidelines for interface design.

6.2 Shared attention in human-computer interface theory

In The Essence of HCI, Christine Faulkner lists several modes of human attention that are applicable to interface design. These modes are subcategories or combinations of certain aspects of focused and shared attention. For each of these categories she presents a set of guidelines for developers of systems that require one type of attention. Faulkner groups attention into four subgroups: Selective attention, focused attention, divided attention and sustained attention. Of these, divided attention is the most applicable to our problem area.

Faulkner (Faulkner 1998) describes divided attention as "several tasks performed simultaneously" and points out people's tendency to try to do more than one thing at once, even when it degrades their performance. She also stresses the importance of practice and the increased likelihood of errors compared to focused attention. Her guidelines for developing systems that require divided attention are listed below:

- Potential sources of information should be limited as far as possible.
- The user should be encouraged to prioritize and prioritizing should be supported.
- The tasks should be as simple as possible.
- The tasks should be dissimilar in terms of input/output/modality so as to reduce the likelihood of confusion between them.

Note the similarity between these guidelines and the criteria for shared attention tasks from cognitive psychology. The first guideline is very general in that it states that input sources should be limited as far as possible. Though it has no direct counterpart in Eysenck's attention factors it is in agreement with his view, as the primary challenge to shared attention systems is complexity. The second guideline is more specific to IT use and interface design. Prioritizing is not a task that is considered by Eysenck's factors, but is very important in some cases. Knowing which of two alarm bells to ignore can be vital knowledge. This becomes even more important in mobile IT as shown in the next section. The last two guidelines are directly analogous to the first and third of Eysenck's criteria respectively. The second of Eysenck's criteria could perhaps have been included here also, but in interface design it can be seen as a special case of the first: If a task is easy to perform, it is probably easy to learn as well.

While these guidelines and Eysenck's criteria are not the same, it is apparent that the general concepts introduced in the last section do have some merit from an HCI standpoint. However, these guidelines are dealing with interface design in a general fashion. Mobile IT poses a unique form of shared attention that is investigated in the next section.

6.3 Shared attention in mobile IT

Shared attention takes on a very important role in truly mobile IT. As explained in the mobility chapter, true mobility contrasts with the traditional concept of mobility in that the user is on the move while operating an IT device, rather than merely "setting up shop" in different locales. Because the user is on the move, he has to manoeuvre and interact with the world around him at the same time as he operates his mobile device. This poses a very different problem than merely adding another information source to a GUI. The real world is chaotic, complex and unpredictable (at least compared to interface components) and presents a radically different set of stimuli than the mobile device does. According to the guidelines in the previous sections, the real world could be considered just another task that relies on many different modalities. But several articles that deal with what we have termed true mobility conclude otherwise.

One article that deals with this true mobility and its relation to shared attention is "Using While Moving" by Pascoe et. al. (Pascoe et al. 2000). This article is described in more detail in the interface design chapter as it aims to describe their experiences as user interface design guidelines. But Pascoe et. al.'s treatment of shared attention in a mobile setting is also of interest to us. Though they concentrate on fieldwork: "much of our work is valid for applications that require mobile usage but are outside of the fieldwork arena altogether, e.g., PDA tourist guides" (Pascoe et al. 2000). Pascoe et. al. consider true mobility such an important factor that they redesigned their entire user interface around this concept. Their goal is what they call the "eyes-free interface" where the user has all his senses free to concentrate on his environment, while the mobile device is operated in one hand using tactile input only. Obviously, the real world is more than just another user interface task. True mobility requires a rethink of how a user relates to a computer program.

So how does one apply the theories of shared attention to truly mobile computing? One attempt at this was made by Beck et. al. in their thesis "Metoder til brugbarhedstest af mobile apparater". Here they combine the cognitive theories of attention with Kristoffersen Ljungberg's mobile modalities to create a framework of methods for mobile usability testing. Beck et. al.'s conceptual framework is primarily aimed at practical experiments. In this section we will concentrate on how this framework applies to shared attention.

Like Pascoe et. al. Beck et. al. realizes that manoeuvring in the real world is a complex and dynamic activity. According to Beck et. al. the mobile use-case consists of three tasks:

- The task of interacting with the handheld device.
- The task of remaining mobile.

- The task of navigating the environment.

This distinction between movement and manoeuvring forms the basis of Beck et. al.'s mobility theory. They combine this concept of manoeuvring with cognitive psychology's concept of *automatic processing*, where a task through practice or its inherent simplicity can be performed without taxing the human mental resources measurably (Eysenck 1984, page 56). Beck et. al. divides the manoeuvring tasks into controlled and automatic processes according to their difficulty in a given situation. Combining the different forms of manoeuvring with their three categories of movement: no movement, constant movement and dynamic movement; Beck et. al. differentiate between seven distinct use-situations. Each of these use-situations is categorized as requiring either focused or shared attention. In addition, they are divided according to single- dual- or multi-tasking requirements. Of particular interest is that according to Beck et. al. multi-tasking requires focused attention as the user is rapidly shifting his focus rather than performing two mental tasks at once.

Beck et. al. are depending heavily on Eysenck's cognitive factors. They utilize these theories directly and when combining them with Kristoffersen & Ljungberg's modalities do not change or subtract from them in any way. Of most interest to us is the issue of whether this is the correct choice when combining theories from such disparate fields.

6.4 Shared attention and mobile 3D systems

In the last section we showed examples of how other researchers have interpreted shared attention in a mobile IT setting. But how does this translate to mobile 3D systems?
Unfortunately, the studies we have found on M3D have pretty much ignored the impact and effects of shared attention. This means that we have no direct references and must use our own judgment in applying current theories of shared attention and mobility to M3D systems. In this section we examine the compatibility of the theoretical models of the previous sections when applied to the M3D paradigm.

Psychological factors

Some of Eysenck's psychological factors still hold true for M3D systems: in that they should be simple and easy to learn. The first of these principles is adequately translated to the HCI paradigm by Faulkner. As pointed out earlier, the second can be seen as a special case of the first. In the system design chapter we have decided to keep them apart, partially because a practical experiment would leave little time to practice using the system. However, when it comes to the principle of similarity, drawing upon these rules becomes more difficult. How "different" is using an M3D system and manoeuvring in the real world? In performing both tasks, humans mainly rely on visual input. This similarity points to the main source of interference between the two activities: Both the system and the real world have to occupy the visual field at the same time. Compensating for this represents a major challenge for M3D systems, and one we have found no clear-cut solution for in literature. We have to look to other guidelines and methods to minimize the impact of each upon the other.

HCI guidelines

The design guidelines of Faulkner are more specific, in that they are aimed at systems developers. The first guideline states that "Potential sources of information should be limited as far as possible". In an M3D system this can be seen as limiting or removing other GUI elements than the 3D world. This makes sense in that the usefulness of these elements is degraded by their distracting effect. In essence, if you have to use real-time 3D on the move, don't use other system features at the same time. The second guideline urges prioritizing. This can be taken to mean that the user should be aware of the relative importance of the different tasks. As a default it can be assumed that the real world is considered more important than the M3D system and so takes precedence. This relies mainly on the user's judgment, but choice of design elements may affect this. This relationship is handled separately later in this section.

Mobile IT

How do M3D systems differ from the systems discussed in the previous section? What makes 3D special? Perhaps the most obvious difference lies in that in contrast to Pascoe et. al. we are bound to an exclusively visual medium. The eyes-free interface makes no sense for a 3D system. It is clear that an M3D system can not rely on this approach for solving the shared attention problem. However, some of the other methods Pascoe et. al. propose are perhaps more appropriate. Examples include one-handed operation and layered menues.

In contrast to Pascoe et. al. Beck et. al. do not provide any guidelines for how to minimize the attention demanded by a handheld system. Instead, their focus lies on creating a theoretical model for shared attention and how to measure the attention that a system demands. In combining the cognitive theories of shared attention with Kristoffersen & Ljungberg's theories of mobility they primarily modify and add to Kristoffersen & Ljungberg's theories. The cognitive theories are not modified, added to or detracted from. To us as informaticists, this seems the obvious choice: We know more about Kristoffersen & Ljungberg's modalities and the field they fit into than we do about Eysenck's criteria for shared attention tasks. Therefore we take these theories at face value, so to speak. We do not propose to alter the theories of another field, but some of their concepts and terms are simply inappropriate in this context. Therefore there are two key areas where we differ from Beck at.

al. in our use of cognitive psychology theories: Firstly, as explained above, using a handheld computer and navigating in the real world are so radically different tasks that comparing them in other capacities than their reliance on visual input becomes inappropriate. The second area where we differ is the distinction between dual- and multi-tasking. This issue is dealt with more thoroughly in the next section.

Dual-tasking vs. multi-tasking

A problem in applying the model of dual-tasking vs. multi-tasking is simply how to distinguish between them. In a theoretical approach, one meets the problem of where to draw the line between very rapid multi-tasking and dual-tasking. This problem is an example of how two competing cognitive models of human thought processes differ in their views. The first model states that the brain acts like a central processor that handles a single problem at any one time and what we see as dual tasking is actually very rapid switching between thought processes. According to this view, there can be no such thing as dual-tasking. The competing view holds that the human mind is capable of handling multiple simultaneous processes, allowing for true dual-tasking. These two models are known respectively as the central-capacity and multiple-resources model (Eysenck 1984, page 64-68).

This problem becomes even more pronounced when applied to a practical experiment. How does one measure another person's mental processes? Beck et. al. (Beck et al. 2002) use their conceptual framework to extrapolate whether the subject uses dual- or multi-tasking from the user's movement and manoeuvring: If the user applies conscious effort to manoeuvring he is said to be dual-tasking (that is, he applies conscious effort to two tasks simultaneously). If he at times merely puts one foot in front of the other he is multi-tasking (the user rapidly alternates between concentrating on the system and manoeuvring). This follows from how Beck et. al. distinguishes between movement and manoeuvring. Note that according to this view, it is possible for mobile users to maintain continuous motion while multi-tasking.

This conflict between the central-capacity and the multiple-resources models relates to the cognitive factor of practice and experience. It is difficult to tell whether a subject in time becomes better at mentally juggling two tasks or recognizes the problems more quickly and so becomes quicker in re-focusing his attention. Several influential studies have suffered from this difficulty, (Spelke et al. 1976), as referred to in (Eysenck 1984, page 61-62).

There is no consensus on which of the central-capacity and multiple-resources models are correct, or even if any of them is the "correct" model (Eysenck 1984). In this thesis, the issue is only relevant when it comes to distinguishing between dual- and multi-tasking. In the experiment design chapter we argue that due to our high level of abstraction it becomes very difficult to distinguish between when the subject applies dual- or multi-tasking. However, these two concepts describe two methods of applying attention that can have implications for system design. Consider a scenario where the user switches between standing still for a few seconds concentrating fully on the device and walking for a short while and ignoring the device. In this *stop-start paradigm* the user can be said to be multitasking, even though the intervals are several seconds long. If, in contrast, the user maintains his motion while interacting with the device, this can be called the *continuous-motion paradigm*. Each of these use patterns puts different demands on the user interface. In the stop-start paradigm, the user has more attention directed on the device and so can be expected to e.g. be capable of entering more fine-grained input. If the user is truly dual-tasking, the input would conceivably be less accurate, but the user would have more time in which to enter them.

In our design, we do not favour any of these use-patterns, but instead we hope to see which, if any, the users prefer. Therefore, we will retain the distinction between dual- and multi-tasking, except that we will apply this in slightly broader terms than Beck et. al. In the discussion chapter, we distinguish between the two as follows: If a user employs the *continuous-motion paradigm* he is said to be dual-tasking. If the user employs the *stop-start paradigm* he is considered to use multi-tasking. This difference between our definition and that of Beck et. al. stems from our simplified view of walking (i.e. walking and manoeuvring can be seen as parts of the same “task”.)

Self-regulation of shared attention

One argument that warrants closer scrutiny is the ability of human beings to regulate their own attention. This ability is in evidence in everyday life: A person watching television wrenches his eyes away at his spouse’s request for conversation. The passengers on a bus remain studiously preoccupied with their own internal thoughts while a drunk is singing loudly. This "attention direction" even appears to be automatic to certain degree. A person talking in a mobile phone ignores it when prompted by the surroundings to focus his attention elsewhere. In cognitive psychology, the commonly held view is that the dominant factor that determines the content of attention is human choice and direction (Eysenck 1984). However, it is also recognized that some stimuli draw attention to themselves. This is particularly true of stimuli that are novel, incongruous or surprising.

It is known from human-computer interface theory that it is possible to direct the user’s attention within a user interface (Faulkner 1998). It has even been shown that the user’s attention can be "trapped", that is that the user focuses on a part of the interface that does not warrant attention. From everyday life we know that stimuli that are sudden and stand out from the background distracts and draws attention to itself. This sometimes leads us to miss more important but less obvious stimuli.

Clearly, the stimuli from a handheld device can have an impact, despite of the ability of human beings to direct their own attention. By following the criteria and guidelines of shared attention theories, this impact can be lessened. But perhaps more importantly: it will be easier for the user to discern which stimuli are most important. In the mobile use-case, Faulkner's prioritizing becomes a matter of when to ignore the handheld system. In fact, we as designers must rely on the user's self-regulation of attention to ensure that he will ignore the system in situations where it would be impractical or even dangerous not to. But since the device does play a role, even if it is a small one, proper design has the capacity to aid the user's attention direction rather than hinder it.

Attention and control

An important application of the above attention theories is the relation between user attention and expected quality of input. As pointed out by Eysenck and Faulkner, the performance of users decreases in shared-attention scenarios. If the users have less attention focused on the device, it can be assumed that their quality of input will decline. That is: they will likely make more mistakes, their input will be less deliberate and therefore less accurate and input will be slower and more erratic. If one considers attention to be a single mental "resource", it follows that the less attention given to the device, the more likely errors become. The quality of input and level of attention become linked. Inversely, it can be said that the more involved and detailed input the system demands, the greater attention it requires. In the Interaction Design chapter we investigate what this means for user interface design in M3D systems.

Summary

In this chapter we introduced the models of shared attention from cognitive psychology. Moving progressively closer to our problem area, we then examined Faulkner's guidelines for shared attention in HCI design. Showcasing two examples of how shared attention is treated in mobile IT, we concluded that the task of manoeuvring in the real world is so complex that it can not be adequately compared with other system tasks. The last section of this chapter dealt with how to apply the lessons from the previous sections to an M3D system. In conclusion, all three treaties of shared attention contribute to our understanding of the subject of shared attention. However, for an initial effort at a practical experiment these models are too fine-grained.

For the purposes of this experiment we will therefore use a simplified model of human attention, inspired by Jameson et. al.'s concept of *resource limitations* (Jameson et al. 1999). Jameson et. al. apply the term primarily to the user's time and memory constraints, bur they acknowledge that there are many aspects of user cognition that face similar limitations. The user's attention can be seen as one such aspect. Viewing user attention as a single resource with a finite capacity is sufficient for our case. This allows us to skip a treatment of micro-level internal processes in the user, while applying the guidelines for shared attention provided by the above theories.

Chapter 7

Design rationale

IN the three latter chapters we have tried to establish a theoretical frame for mobile IT use, with 3D visualisation in focus. From this background we now seek to establish some principles or requirements for our test bed system that aim to lessen the users memory load and that take use of 3D technology. The scenario chosen for the test bed system is a navigational aid (described in the next chapter).

7.1 System Requirements

During the course of our pre-study, it became clear that the main focus of our system design would have to be compensating for shared attention. To do this, our test system needed to fulfil certain criteria. These criteria developed into the set of system requirements that we describe below. In contrast to the general design guidelines described in the introduction chapter (section 1.4.2), these requirements focus on the usability of the system, not the experiment setup.

Below, the four requirements are described. For each requirement we describe the following properties:

- Description
- Theoretical basis
- What design means we implemented to fulfil the requirement
- How we seek to find evidence regarding relevance of requirement in light of shared attention.

7.1.1 System must aid user in dynamic setting

This system requirement may seem obvious, but is none the less essential. If a user is to be able to use a 3D system while on the move, the system must be of use to him in such a setting (Beck et al. 2002). There is no need to optimize a system for 3D turbine design for use while a person is walking, since such a system would simply not be useful to a person who is walking down a city street. There must be a reason for a mobile person to use the system. A good example of this is the Fieldworker system, (Pascoe & Ryan 1999). Here the system is designed as an aid to biologists who record sightings of animals in African national parks.
However, one of the key features of mobility is the *dynamic* use model, as pointed out in the chapter on mobility. This means that the way the user interacts with the device changes according to the situation. As the use-mode changes, the demands on the user interface changes also. It would seem that making a dynamic user interface would enable designers to optimize the way the user interacts with the device to the current use-mode. As described in the Interaction Design chapter, Pascoe et. al. have an interesting variant of this. They provide the biologists with a standard pen-and-touchscreen interface for use during breaks in observation and a reduced, “eyes-free” interface for use during observation. The concept of the dynamic user interface holds much promise, but applying it to the navigational system described in this thesis posed problems.

Perhaps most importantly, as pointed out in the introduction we did not have a clear idea of how the subjects would react to the system. In other words, it was unclear what use-mode they would fit into. Designing for specific use-modes would force our concepts on the users in a very direct fashion. As we were more interested in finding out what type of use the subjects would adopt, we chose to use a single GUI design. To present the navigation data to the user we chose several different 3D manoeuvring functions. As described in the chapter 8, these were implemented as VRML “Viewpoints”. These functions gave the user varying degrees of control over his movements in the 3D world. But the more control the user gets, the more complicated movement becomes. As pointed out by Faulkner (Faulkner 1998), the more complicated a task is, the more difficult it is to perform simultaneously with other tasks. We therefore implemented the system so that we enabled several ways of manoeuvring in the 3D space, according to the users preferences and more importantly the use-mode. The most complex form of movement we called “Free” and could be triggered by choosing the “Free-viewpoint” option in the Cortona menu-system. In contrast, the “Walkthrough” viewpoint made the user follow a set path towards his destination. In the experiment we hope to see when and why the subjects used the different navigation possibilities. If we can find a pattern, we can perhaps draw conclusions as to whether our 3D system did indeed offer such a dynamic use-mode or not. We may also strengthen the argument that a truly mobile system *should* offer such possibilities.

Shneidermans “visual seeking mantra” (Shneiderman 1996) can adopt dynamic properties if one can enable a dynamic transition between overview, zoom, filter, and details in the system. For instance using the system to get “detailed” information may require a different use-mode than getting an overview (in our system: free-viewpoint provides detail wheras birdseye-viewpoint provides an overview). This principle therefore inspired us defining this requirement.

7.1.2 System must provide easy and fast navigation

Regardless of what precise tasks our system was to perform, the user would have to navigate a 3D world, preferably while remaining mobile. This posed serious challenges, not least because our chosen platform, the PDA, is not designed with 3D navigation in mind. The most popular user interface that PDA’s commonly offer is a combination of a stylus and a touch-screen. This approach has the advantage of utilizing the available control space very efficiently and giving the user a very direct means of control, but has several drawbacks to a mobile 3D system. Firstly, using only the stylus to control movement means that turning, speed regulation and direction are controlled using a single means of input. This makes 3D navigation on a PDA difficult even for trained users. Secondly, the user’s hand and stylus can obscure a significant portion of the screen. This is especially noticeable when the user wishes to move at speed through the VRML world, since speed is controlled by the distance the stylus is dragged across the screen. It also makes turning to the left difficult for right handed users and vice versa. Thirdly, the stylus can be moved in any direction along the 2-dimensional screen, but how does one signal that one wishes to move "into" the screen?

The Cortona viewer takes advantage of the different VRML navigational modes to allow the user to move relatively freely when using a stylus. Basically, each navigation mode allows movement along two of three axis, with the third being inactive (the VRML navigation modes are discussed in more detail in section 8.5). This approach has the advantage of allowing 3-dimensional movement without resorting to using the hardware buttons on the PDA. Using these buttons would mean that the user would have to use both hands to navigate, which would be difficult when using one hand to hold the device, or having to switch between the stylus and the PDA buttons. The drawbacks of the Cortona viewer’s navigation is that the user is restricted to moving along only two axis at the same time and that the user has to spend quite a lot of time switching between navigation modes. The first problem is not strongly felt as long as the user moves along the ground plane of the 3D world, as there is no need for him to "fly" along the Y axis while simulating walking. But if the user wishes for a birds-eye view or just change the direction he is viewing, by for example looking upwards in the 3D world, he will have to change user modes at least twice.

We found that these drawbacks seriously reduced the efficiency of test systems, so we devised a simple solution: limited automation of navigation. On the surface, this goes against some of the primary advantages of 3D systems, namely the nearly unlimited viewpoints and free navigation and control. Having the user select from predefined paths instead of navigating the world himself negates these advantages and places a huge burden on the developer who has to design these paths. However, this is not necessarily the case. Path design can in fact be performed by the computer: By using an appropriate algorithm, the computer can calculate the optimum path from a set of predefined instructions. See (Duvaas 2002) for an example of such a system. The workload of the designer is increased, but the result can be a significantly enhanced user experience. In accordance with our principle of providing multiple choices for the user, we decided to implement several different modes of navigation, each with a slightly different level and kind of automation. Seeing how users valued freedom opposed to convenience in 3D navigation would be an interesting part of our experiment.

7.1.3 System must address fast recognition - primarily by realism

We saw realism as the key to making the connection between the 3D world and reality. We did not know in advance what features people would find distinctive in the real world, but we suspected that these would vary between individuals, so we aimed to replicate as many details as possible. We hoped that our interviews would shed light on what kind of features and details were necessary to a 3D map and what features could be discarded. Our main tool in creating a realistic world was basing the geometry on aerial and satellite images and using photographs as the basis for our textures. Unfortunately, reality can become quite confusing sometimes, and we found we could not adhere strictly to reality in all cases. The most major changes we implemented was simplifying complex objects that could cause confusion (removing minor architectural peculiarities that would impede movement in the 3D world) and exaggerating the traits that make objects unique and set them visually apart from others (making a metallic statue a little bit more metallic than the photograph shows). However, realism remained our guiding principle and we sought to emulate reality as closely as possible in our 3D world.

By far the majority of the simplifications we made came about by the limitations set by the handheld terminal and the available software. The changes we made were usually because the low resolution of the model made some characteristics difficult to see or even disappear completely. Some images also had to be enhanced by making them lighter, since we only had limited quality digital cameras available to us. The discrepancy between the photographs we took and how the object actually "looked" was sometimes quite striking. In some cases it was impossible to make out such simple details as windows. In the most extreme cases we were forced to photograph tiny parts of a building or feature and build up a complete texture from this. On the largest objects, such as the ground plane and the lawns, we had to create tiled textures to save space. We go into more detail on how the world was built in chapter 8.

In the experiment we hope to see if the subjects were able to recognize and couple the 3D model of campus with their surroundings without compromising in a great extent their given tasks.

7.1.4 System's GUI must be consistent and be within the realm of 3D when possible

A Graphic User Interface is an essential part of most systems, and ours is no exception. The Cortona viewer includes its own GUI that lets the user choose viewpoints, speed and movement type (see illustration 8.2 - lower menu bar). We could implement some of the functionality we required through this interface, but not all. Particularly, we decided to provide the user with a compass to aid navigation and textual information about his tasks. These things required a second set of GUI elements. Traditionally, the GUI elements are displayed in 2D and either overlaid or located along the edges of the user's field of vision. In mobile 3D applications the GUI presents special problems because: Handheld terminals have small screens, meaning that a GUI will take valuable space away from the 3D world itself. A GUI requires added attention from the user, compounding the shared attention problem.

There are two types of 2D interfaces that are commonly used in 3D applications: dashboards and Head Up Displays or HUDs. The dashboard type that is used by Cortona displays the GUI elements on a background reminiscent of the dashboard layout of vehicles. The dashboard and its GUI elements are often located along the bottom of the user's field of vision, as is the case in Cortona. In contrast, a HUD displays the GUI elements overlaid on the user's field of vision. The term HUD originally comes from the see-through part of the dashboard on fighter planes, where it is held to enhance a pilots ability to utilize both instrument information and environmental information simultaneously (Lauber et al. 1982). The premotor theory of attention holds that each shift in attention gives rise to an eye movement. The HUD design seeks to minimize the impact of attention shift by minimizing the associated eye movements while limiting the user's field of vision as little as possible. As unobtrusive as it may be, a HUD still requires the user to shift his attention from the 3D world to the GUI (McCann et al. 1993). A HUD also presents its own problems: attention can be "trapped" by a HUD by making the user pay more attention to the HUD than the world around him (Clark 1999). This can happen when the HUD obscures a vital change in the real world, or when changes in the HUD leads the user to ignore changes in the real world. A HUD design still seems preferable to us when one considers the challenges listed above: A HUD takes up less space on-screen since its elements are not connected by a dashboard; and it becomes easier for the user to shift his attention between the interface and the 3D

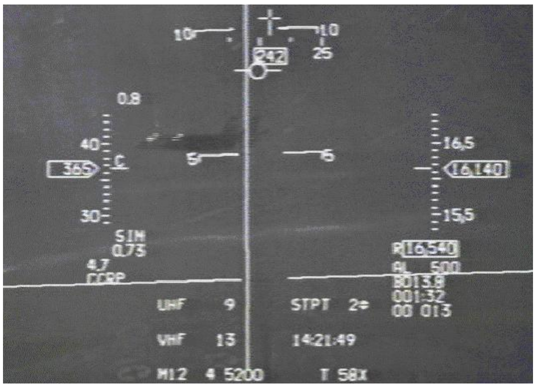

Figure 7.1: Illustration: The view from the cockpit of an F-16 fighter plane shows the HUD display. The lines in the centre shows the plane's direction and elevation, while the other elements display vital information such as airspeed and altitude. Newer versions of fighter pilot HUD's are mounted on the pilots' helmets instead of on the dashboard.

world.

We were facing some technical challenges when designing a GUI, whether a dashboard or a HUD. As explained in section 4.8, we had no way of creating a true 2D interface in our system due to limitations in VRML. The only method we had available to create a dashboard or HUD display would have been to project the GUI elements on a "plane" that is locked in place in front of the viewer. This is shown schematically in illustration 7.2 A. Traditionally, the interface buttons, here labelled 1, 2, 3, 4 and Enter, are displayed on a "dashboard" shown here in green. This is usually overlaid on top of the 3D world, like in the Cortona interface shown

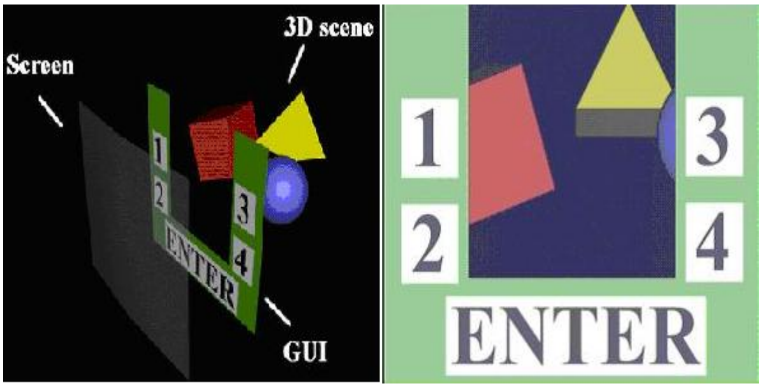

Figure 7.2: Illustration: A) This image shows a schematic overview of a HUD-like GUI implemented in VRML. B) This is how the same HUD would appear from the perspective of a user. The dashboard and the buttons obscure part of the screen and distracts the user from the 3D world itself. This is an exaggerated example, but illustrates the problems posed by a "dashboard style" GUI.

in illustration 7.2 B, but in our VRML system, we would have to make this type of GUI as a separate 3D object inside the scene. This would have been a less than ideal solution as the various GUI elements would have required a complex set of 3D objects with their own scripts to govern their movements relative to the user, or alternatively, a very resource-intensive set of animated textures on the GUI plane.

This left us with two problems when designing our interface: the GUI required added attention from the user, and creating a 2D GUI would be a cumbersome process that would result in a slower framerate. A possible solution to these problems came from Augmented Reality:

Rather than displaying the GUI elements as 2D projections on the user's field of vision we decided to integrate them directly into our 3D world.

This would free us somewhat from the limitations of VRML by allowing some of the GUI elements to be coded as stand alone 3D objects. Also, according to premotor theory, making the GUI information an inherent part of the 3D world would make the necessary eye-motion of the user even less than those required by a HUD. According to AR theory, embedding the computer-supplied information into the environment provides the user with a single coherent experience. In addition, a 3D GUI shares many of the advantages of other 3D content, that were very useful to us. Of special importance to us was the fact that a 3D GUI element can take up less space than a 2D equivalent. For example, consider the 3D compass arrow. Most of the time, the arrow will be aligned with the ground plane, meaning the user looks at it from an almost flat perspective, as opposed to a 2D design where the arrow would be displayed full size all the time. A 2D arrow would constantly require the amount of screen space that a 3D arrow only takes up in special cases. The 3D arrow takes up a lot less space, but conveys the same information as a 2D design, or perhaps even more. Displaying the elements in 3D can add information by itself, as again demonstrated by the 3D compass.

A 3D GUI works best when the number of elements remain small. If the number becomes too large, they will crowd the screen and make manoeuvering and recognition difficult. Fortunately, our design called for just a few GUI elements, such as a compass, an information cube, a lighted path and an information screen. Most of the time only the compass and information cube would be visible, allowing us to place the compass in the lower left side of the screen and the information cube on the lower right side of the screen. These advantages come at a cost, however. Introducing perspective can make markings on the GUI elements difficult to see. This can be somewhat remedied by making sure the markings are very simple and easily readable from most angles, though this is not always possible. Similarly, the gains in design space by introducing perspective can be marginal and require special attention to utilize fully. We wanted to make the GUI seem like an integrated part of the presentation, but we also wanted to signal to the user that the GUI

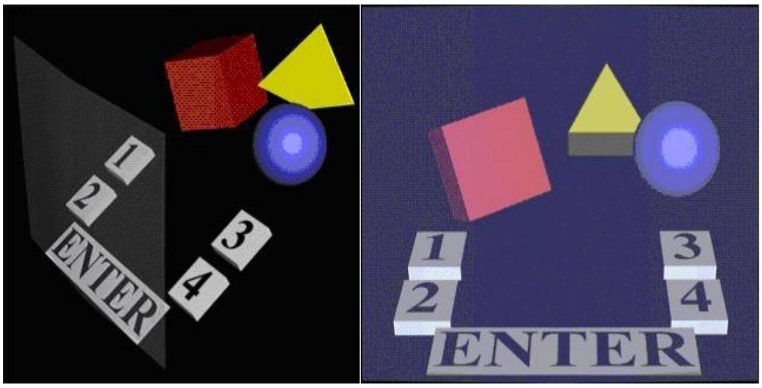

Figure 7.3: Illustration: An example of a 3D GUI. Here the buttons from the previous example have been turned into 3D objects that are part of the scene. Not only do they take up less space due to perspective, but they appear as part of the scene, not an extraneous element. Note that this example exaggerates the size of the GUI elements to make the effect of perspective more obvious.

elements were not something he could expect to see in the real world. This might seem inconsequential, but we wanted to make absolutely sure that they were not confused with statuary or other art pieces, they were made to look distinctive. We did this in two ways: Firstly, contrasting to the photo-based textures of the rest of the world, *the GUI elements had bright, cartoonish textures composed of one or two colours*. Secondly, *the GUI elements were in motion relative to the rest of the 3D world*. Throughout the user's movements, they remained fixed in space in front of him. The GUI elements differed from the objects corresponding to real world counterparts in that they rotated over time. The compass reoriented itself as the user moved, and the infocube span slowly around its centre axis. The other GUI elements, the infoscreen and the lighted paths, were deemed to be obvious enough that animating them would be superfluous. We hoped that this would make the GUI elements obvious without making them too attention-grabbing.

7.2 Hidden annotations

Annotations are an important feature of Augmented Reality. In fact, they are so essential that AR is sometimes known as Annotated Reality. In AR, annotations are pieces of computer supplied information added to the real world. In our system, we planned to use a variation of annotations to tie information to objects inside the 3D world. Hopefully, the user would be able to recognize the objects from the real world and connect the supplied information to them. Unfortunately, presenting annotations presents special problems on a PDA platform. These are mainly caused by the PDA's small screen and combination screen/interface. Perhaps the greatest challenge lies in that readable annotations have to take up a large part of the screen, obscuring much of the 3D world. This problem is compounded by the mobile setting. The annotations must be made very large and high-contrast to be readable in conditions of glare, common in an outdoors environment. They must also be legible while the user is on the move and has little attention to spare to decipher minute lettering. In this case, only one or two meaningful annotations could be shown at the time without obscuring the 3D world completely.

We first thought to compensate for these problems by using expandable annotations. Each object in the 3D world about which the system supplied information would have an attached symbol. By clicking the symbol, the system would display with the full annotation about the object. This solution seemed very well suited to the PDA's touchscreen interface as users could quickly and accurately tap the object they wanted information on. In practice, this method was less successful. Trial versions of the system showed that this method had two major flaws: Firstly unless a CPU-intensive script was used the symbols would change in size because of perspective and be obscured by buildings in between. This meant that there was no guarantee, even if the user could see the building quite clearly, that he would manage to click the annotation symbol. Using a script to compensate for this would be far too costly in terms of slowing the entire system down. Secondly, as the number of symbols increased it became more and more difficult to navigate the 3D world without accidentally clicking on an annotation symbol. Using partially automated navigation methods compensated only partially for this, as the user still needed the option to move freely. Using a script to limit the activation of the symbols until the user was within a certain distance made information retrieval cumbersome and slowed the system down considerably.

Looking for an alternative solution, we found that we could combine the annotations with the existing GUI setup. We had already decided to include an "infocube" in the lower right corner of the screen. By clicking the infocube, the user would receive a full-screen textual description of his task. In a full system, the user would be able to select a "target" in the 3D world which he wanted to find. Selecting the target could be by inputting the address, clicking it following a special command or be designated automatically by the system. The user would also have at his disposal a scrolling menu with the names of all the buildings and other objects of interest in the 3D world. The name of the building or object currently designated as the "target" would be highlighted inside the menu. By clicking the infocube, the user would receive a full-screen textual annotation about the target. This would allow us to hide the annotations effectively while limiting the number of steps required to read the annotations to two. Due to time constraints we were unable to implement this feature, but we could test its effectiveness by seeing how users responded to the task descriptions, which in effect became part of the annotations in our system.

Chapter 8

System description

IN this chapter we will describe how the system is developed and designed. The system includes VRML files, image files, a VRML-browser application and a PDA with Win CE 3.0 OS. These components together creates the environement for a realtime 3D virtual world. Recall section 4.8 for an introduction to the VRML standard.

The test bed application takes the form of a set of four VRML-files. These are located in a single directory and displayed to the subject after each task's completion, allowing the subject to select the next task by clicking on the file. This was done because the Pocket Cortona browser does not have proper support for linking between VRML files. Since the core and largest part of the system is a 3D model of Blindern campus, we used a single set of texture images that was used by all four .WRL files to save space. The images have been compressed as much as possible while still retaining a sufficient level of detail. One of the most obvious and persistent technical problems was the long load time of the .WRL files. We found that this was caused mainly by the Pocket Cortona's handling of Text nodes, but fortunately we were able to design the experiment so that load time would not affect the experiment unduly.

8.1 The translation from geodata to 3D objects

The initial phase of the system development was to translate the geographic terrain and building elements from campus to a digital virtual world on the PPC (Pocket PC). Technologies for automatic generation of 3D landscape models are already available. Though we created our campus model based on maps and architectural

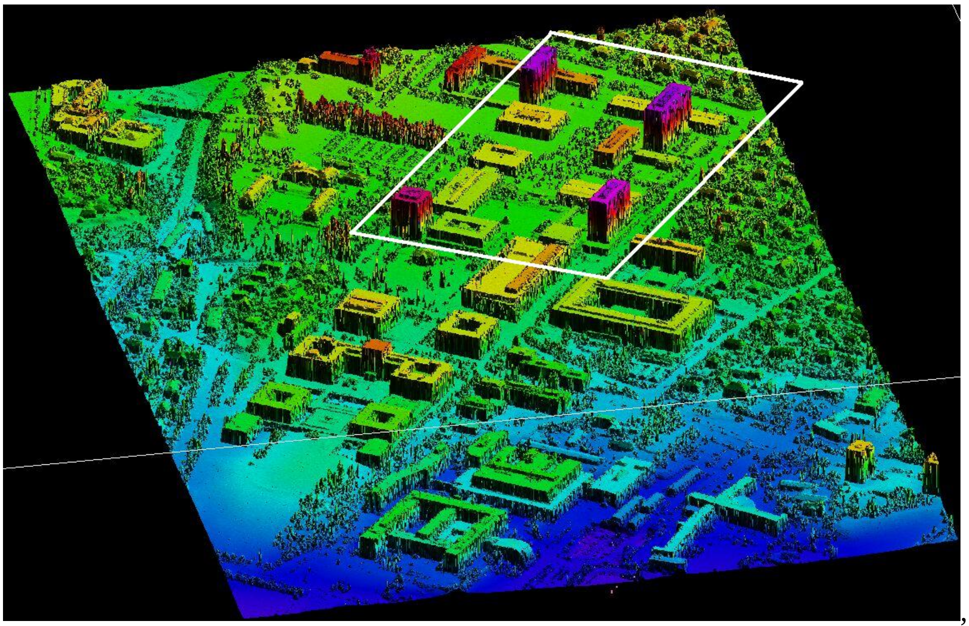
,

Figure 8.1: The Blindern campus as seen with a MicroStation GeoOutlook laser terrain mapper mounted in an airplane. The area included in the 3D model is shown inside the white rectangle. Image graciously provided by Fotonor (Fotonor AS 2002).

information, automatic terrain data generation is a possibilty. As an example, figure 8.1 shows a 3D model of Blindern campus created from data supplied by a laser terrain mapper.

We stribed to do the manual translation as accurate as possible in order to present the 3D world realistic to the user. The working method was to explore the campus area by taking photos and computing distances. We then redesigned the geodata on a computer using the tools; 3D Studio Max, Cozmo Worlds and Photoshop. After exporting the 3D world to VRML-syntax, we altered the code with a text editor to tweak the performance and to optimize for interactions. We eliminated designing detailed environments and objects we found too complicated or irrelevant for our purpose. We did not want to make the model too detailed because we are in the opinion that details that dont affect the realism or have no purpose in our world (no functions) can be left out without compromising the intention of the experiment. We had to make the geometry (3D surfaces) as simple as possible because the number of surfaces strongly affects the framerate - which is critical for a real-time experience. Furthermore: mobile terminals today are very hardware constrained in performance thus making this criteria even more vital.

Some of the most important objects in a 3D world is objects that can be used as landmarks (Lynch 1960) Kevin Lynch argues that the image of a place is important to how people understand and use their environment. The result of a project described in the article “A 3D city info for mobile users” (Rakkolainen et al. 2000) explains the importance of 3D landmarks:

> *Our results show that search and visualization of location-based information of a city becomes more intuitive with life-like 3D. The users recognize landmarks and find the route in cities easier with the 3D model than with a symbolic 2D map.*

We put a lot of effort into modelling objects so that their spatial cognitive relationship enhance the users orientation ability. some specific objects worked as independent local landmarks (e. g. the Air statue[1], while others objects combined (e.g. large buildings adjacent to each other) worked as landmarks at a larger scale. The method we used to make landmarks stand out was to slightly exaggerate the object size, increase texture quality and to make the object even more visible by contrasting the background to a bright blue colour (the sky colour emerges from dark blue, 12 o’clock, to light blue at the horizon using the SkyColor and SkyAngle nodes in VRML).

8.1.1 Simplified representations

The level of optimization required meant that we had to make some difficult decisions on what features to include and how to deal with others. The main buildings were most easily represented by variations of boxes (for an example of a VRML box node recall section 4.8.2) with textures showing details like verandas and stairs. The details from the photograph were generally enough to make the impression of three dimensional architecture. Early tests showed that even if this method became obvious, and the user realised that structural elements were merely photographs, it was easy to recognise the objects from their real-world counterparts. While the buildings could be dealt with in this way, certain other features of the campus needed more careful modelling.

The statue visible in figure 8.4 is a case in point. In the real world it is a very complex shape, with silver tubes, spikes and plates intertwined. In our model, however, it was represented as a two-sided, semi-transparent texture on a single "face" (recall figure 3.1 for a description of 3D faces), essentially making the statue into a 2D "billboard". This 2D plate was placed on a pedestal formed from five faces (the bottom one would be invisible and so was deleted) narrowing towards the top. The pedestal gave the statue a sense of being "three-dimensional" and having a volume, despite the fact that the statue itself was a 2D plate. The statue’s texture was based on a photograph of the statue taken from the south side. This side was chosen because it showed the statue’s most distinguishing features: a hole near the top, and a sharp spike pointing towards the Frederikke building. Looking at the statue from

---

[1] The Air statue is a statue easily recognizable located at the lower central part of campus. The statue is visible in figure 8.4.

other angles than directly from the north or south in the real world obscures these features somewhat. It was important that these features were retained in the model, as they gave the statue a sense of direction by making it asymetrical. By relying on this photograph and the simplified geometry, we retained the most distinguishing features of the statue while only requiring the VRML browser to render a single face.

The disadvantage of this approach was that the 2D geometry of the statue makes it invisible when viewed directly from the east or directly from the west. Tests showed, however, that in order for the statue to become invisible, or narrow enough from perspective to be unusable, the avatar would have to be placed almost precisely along the statue's east-west alignment. This would rarely be the case, so the effect was considered an acceptable trade-off.

The same method was used to represent the fountain and, with certain modifications, the trees also visible in figure 8.4 behind the air-statue.

A more complex feature was the stairs found around the campus. The campus is constructed on three levels: the main level, containing the square, and the Fredrikke and Akademica buildings. The second level with the Math building and Sophus Lie's Auditorium and the third level with the SV and HF buildings. These levels where connected by stairs and ramps, and several buildings featured additional stairs. These stairs were deemed necessary for realism purposes and representing them with flat surfaces with the stairs "painted on" with textures failed to give the impression of three dimensions. We therefore decided to model three steps for every level. This, combined with the faces that made up the level itself made for a total of five steps per stair, while being restrained to six faces per stair. While the real stairs contained between ten and fifteen steps of varying inclinations, this model managed to convey the impression of stairs without demanding too complex architecture. This also allowed us to save memory capacity when texturing the stairs, as they could be assigned a single shade of gray, and still retain their three-dimesional appearance.

However, this solution presented an unforeseen problem: we found that our lighting scheme (see the "Lighting" section) made the stairs appear as a single, featureless plate, since all the faces of the stairs reflected the same amount of light. This problem also appeared when texturing the target for task 2: the container. As the container object was also uniformaly coloured, the edges of the faces became invisible. The solution we found was very simple: we altered the textures of these objects to include highlights and shadows in appropriate places. This made the geometry apparent, but at the cost of more detailed textures.

The simplified stairs also caused another problem: VRML includes an Avatar object that is normally invisible to the user. Nevertheless, this Avatar has certain characteristics, such as height, whether it is bond to gravity and so forth. We designed the campus model so that the Avatar object was 1.75 meters tall, about average for a human, while the campus and the buildings were to scale. It turned out that simplifying the stairs made them too tall for the avatar to climb. And we could not alter the Avatar's height because that would make the viewpoint too tall. The solution came from the VRML standard itself: by altering the Avatar's knee-length (changing the value of the avatarSize parameter in the WorldInfo node, Appendix A), we could make it take longer strides without altering its overall height. This gave the avatar unrealistically long legs, but since the user would never see his own avatar, this was deemed to be of no consequence.

After the most important objects of the campus area were simplified, scaled, modelled and placed in an coordinate system relative to eachother, we proceeded with texturemapping.

8.2 Mapping the textures

Texture mapping is a technique to wrap an image around a surface. The image can be any size but the filetype is important if transparency is to be used. Use .gif or .png image types for this purpose. In VRML, texturemapping can be done in many ways. The easiest method is to wrap a texture around an object by using the *ImageTexture node* (see section 4.8 for an example). A texture can be applied one sided or two sided simply by using the solid flag and no ambient intensity (overall light reflection) on the surface. This is set up in the objects' node description. In our 3D world we hid backsides of almost every object to decrease the CPU-load and rendering-time. We used a rather complicated texturemapping technique in our system. We combined several textures in one image file and wrapped portions of this image at each surface. The roofs at buildings for example was texturemapped by using one "dark" pixel from the image and multiply it to cover the entire roof (Appendix A). By using this method we beleive that we increased performance by having fewer image files; hence the overall storage size for images was reduced and the VRML-browser had more free memory to work with[2]. On objects other than building objects we used a different texturemapping method. The terrain on campus consists of grass and bricks. One square meter of grass differ unnoticably little from the next square meter. Therefore the obvious choice was to make the texture tileable. A tileable texture is a texture than can be repeated infinitely number of times both in x-axis and y-axis with seamless transition. The textureimage size can therefore be made very small.

The most complex terrain object was the model of the stairs. The stairs combine the lower central part of campus with the upper north part (see figure 8.4) Each step consists of a vertical and horizontal surface. We decided to leave out texturemapping here because mapping the texture to both surfaces (one step consist of a horizontal and vertical surface) made it appear as one plain surface becuase of the lighting decision described in the next section. The final stair design was

---

[2] On the Compaq IPAQ, memory and storage space is the same. All data is stored in memory

therefore untextured and with uniform grey colour.

During the photo sessions at campus we stumbled into an unexpected problem. Taking a photo to be used as texture requires the image to be uniform in shape (rectangle with head-on perspective). You can imagine taking a photo of a tall building standing on the ground next to it. The shape of the building in the photgraph will be thick at the bottom and thin at the top because the top is a longer distance away. Similar problems occurred with wide buildings. A great deal of “cheating” was done to cope with this problem. On some building-textures we took a photo head-on on at the ground level and then copied the first floor with some modifications to act as individual floors up to the roof. On som wide buildings we (if it did not compromise the realism) copied one of the corners and flipped it so that the building texture was symmetric. However this required a lot of work, so we did not go through with it on all building-textures.
We found it somewhat difficult to get the lighting and compression ratio right. On some of the textures we duplicated building bricks and small areas of other buildings that we found of good quality by assembling them as an approximate representation of that building. This was a trade-off between realism and clarity. We chose to make the representation as easily recognizable as possible, at the cost of loosing minute details in building structure and scale. This was considered acceptable mainly because the low resolution of the iPAQ’s screen made such details invisible anyway.
For a general texturemapping description, read the article (Ames 2003).

8.3 Lighting

Lighting in a virtual 3D world is a set of complex calculations with many factors involved. Since we are restricted to use a resource-constrained mobile device with a small CPU capacity we had to simplify the lighting effects as much as possible. Normally, light in a 3D world will reflect from surfaces bouncing a predefined number of times between surfaces. This was to complicated for our small CPU. Instead we configured the light rays (from a strong sun source about 45 degrees on the sky) to only bounce once; from buildings directly to the viewer. This made our world look dark, like dusk time, especially in places where the sun rays does not hit directly. To compensate for this lack of illumination we added artificial luminescence to most of the objects in our world. In VRML this lighting feature is called specularColor.

8.4 Interaction and user interface

VRML focuses mainly on 3D models and we found the standard difficult for designing a good user interface and setting up interactions. The 3D world created was adequate as a simple model of campus, but since we wanted to use it in a scenario, as a handheld guide to users, we needed to set up some user interactions and menus. One powerful feature of VRML is the scripting abilities. Scripting in VRML is the ability to change a 3D world dynamically by monitoring user (or other events like timer, position etc) and set up an trigger->act relationship. Often VRML scripts is used within an Internet page for interrelated communications between the HTML and the VRML languages. As explained earlier (section 4.8.3), we could not use HTML in our system and therfore we had to find another solution; scripting within VRML only. This gave us the means to handle advanced interactions, but the system lacks the opportunity to handle text input. Two different interaction methods were used; pullup menus, and interactive 3D objects.

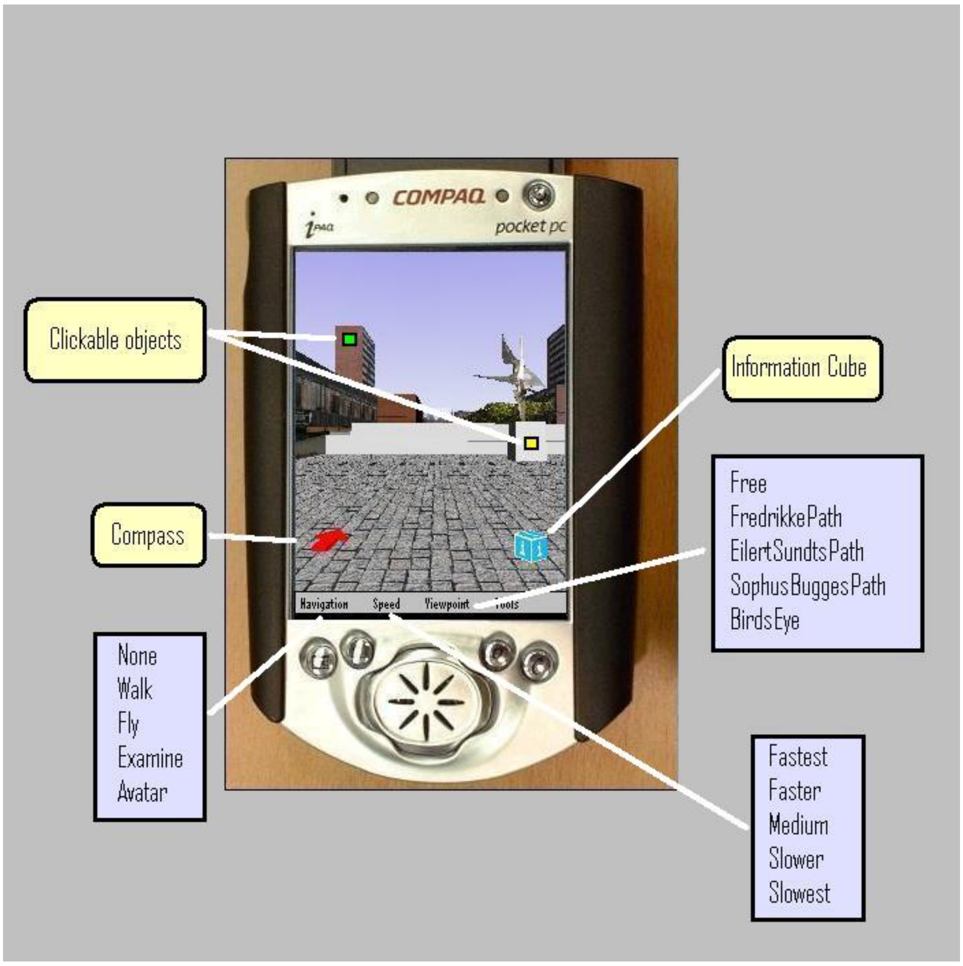

Figure 8.2: Overview of the GUI components - screenshot: The clickable objects (showed as a green and a yellow square) were not a part of the experiment system. We deemed this feature to be of little use, because of poor handeling of “links” in the Pocket Cortona VRML browser.

8.4.1 Pullup menus

Within the Pocket Cortona VRML browser application (as with most VRML browsers) is a set of interaction menus predefined by the VRML standard. This menu choices can be altered and new ones can be added. The figure above shows an example of how we set up a menu for predefined routes within the 3D world (marked in figure 8.2 as blue squares at the bottom). The different viewpoints used in our test bed system is described in section 9.5.

8.4.2 Interactive 3D objects

Besides automatically configuring preset paths and views we decided to present built-in “3D tools” to help a user navigate the system. This 3D interface were as much as possible presented as an integral part of the 3D environment. This menusystem, in contrast to the built-in GUI of the Cortona viewer (the popup menues), does not acquire the user to switch from the 3D world to an external 2D set of text-menus but instead takes advantage of augmented reality principles to merge the menusystem with the 3D world. One example of this is that the user can move freely around and at the same time read and chose from menus. The interface "menu" is very simple, consisting of a navigational aid (3D compass) and a Help-function (information cube).

3D compass

The 3D compass takes the form of a 3D red arrow located in the lower left corner of the screen. The arrow remains stationary relative to the user, but changes its orientation to always point due north parallel to the ground level. This means that it is easy for the user to discern north even when facing straight up or straight down. The compass is a modified version of a 3D compass made by David Ott at the University of Geneve. The modification we found necessary to work out was to simplify the 3D geometry and to make the compass pointing “north”. The original compass pointed to origo, and we therefore moved the origo a long distance by the z axis to simulate the north pole. The code for this compass can be reviewed in Appendix A.

3D infocube

We needed a way to provide the user with the necessary information required to perform each task, which took up minimal screen space and was intuitive to use. We ended up with a rotating 3D cube, bearing the blue and white "i" commonly signifying information on all six sides. The infocube is at startup placed as a rotating box in the lower right corner of the screen. By clicking the cube with the stylus, a blue information screen with white text fills most of the screen and the information cube moves to the upper left corner of the information screen. Clicking the information cube when the information screen is visible moves the information

Task 0

Eksperimenter og bli kjent med systemet

Figure 8.3: Infocube when activated. The text reads “Experiment and familiarise yourself with the system.”

cube back to its original position and makes the information screen invisible again. We were forced to use this approach because the PDA screen is so small that in order for the text to remain easily legible, especially while the user is on the move, it has to fill most of the visible area. The downside is that the information screen blocks most of the view to the 3D model and forces the user to re-orient himself when the screen is removed. In order to avoid this, we made the screen transparent enough to allow the user to see the 3D environment while still being able to read the information text. In fact it is possible for a user to navigate the 3D world while the infocube is activated.

The infocube is coded in a special way. It gives the illusion that when clicked the text-board appears out of nothing. But in fact what the user cannot see is a hidden object that is placed just behind the avatar relative to the avatars position. When the infocube box is touched this board emerges from behind to reposition itself right infront of the user and the infocube itself disapeares (is repositioned behind the avatar). The board has a copy of the infocube in its upper left corner that can be used to “minimize” the board (set the board back to its original position).

8.5 Navigation

We implemented several modes of navigation through the VRML concept of viewpoints. This allowed the subjects to use the VRML taskbar (see figure 8.2) already present on the bottom of the screen to control the navigation parameters. We provided the users with different predefined modes of navigation, each implemented by creating a VRML viewpoint and customizing its constraints. VRML allows several movement types that can be predefined and/or determined by the user. We used two predefined movement types: Walk for viewpoints that simulated the users movement, and Examine for viewpoints that did not. When using the Walk mode, the user moves forward by dragging the stylus upwards on the screen, and downwards to move backwards. By dragging left and right the user swings the viewpoint in the corresponding direction. Examine works a little differently: Dragging the stylus in any direction moves the viewpoint along the inside of an imaginative sphere around the centre of the viewpoint. Though the users were given a brief introduction to the different movement types, we did not expect them to make much use of this feature, as it is relatively advanced and the predefined movement types were designed to be the optimal for the given viewpoints. We were concerned that using the touch screen/stylus combination would be difficult in 3D, so we did not expect the users to make full use of this feature and instead focus on the automated viewpoints. What follows is a brief list of the different viewpoints and their peculiarities.

Free

At system load-up the initial viewpoint is the Free viewpoint. This viewpoint always starts out being placed next to the Air statue, an easily recognizable landmark, and facing north. The Free viewpoint is bound by gravity and uses the "Walk" predefined movement type to simulate the user ambling about Blindern. Free only allows user controlled movements, and so is rather difficult to master.

Groundpath

Quite similar to Free, but providing a path that the user can follow to his destination. The orientation is changed so that the perspective is relative to the way a user should start walking in order to reach the destination. The path takes the form of a golden, translucent sheen to the pavement that forms into an arrow pointing at the user’s current destination. It is composed by a 3D object which geometry is defined in a *IndexedFaceSet* node. The 3D object, which is generally a flat surface, overlays the ground texture and is semitransparent. The path appears triggered by the viewpoint we call “Groundpath” and if “Birdseye” viewpoint is selected the groundpath object is repositioned (using scripts) to appear at a higher altitude (increases the y-axis value in transform attribute) so that the groundpath looks like a pointer from a birds eye view perspective.

Walkthrough

The Walkthrough moves the users view automatically from the starting position to his destination. We took care to make the animation as smooth and easy to follow as

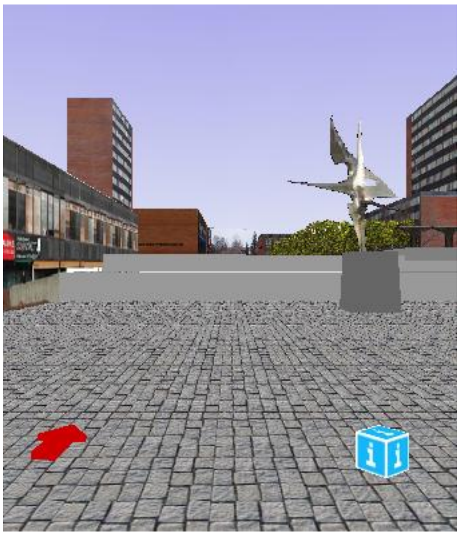

Figure 8.4: Screenshot from free-mode. This is the initial starting viewpoint at each task in the experiment described in the next section. Placed at a base socket, the Air statue can be seen in the middle right. The stairs connecting lower central campus with upper north area can be seen as a grey horizontal bar just behind the Air statue.

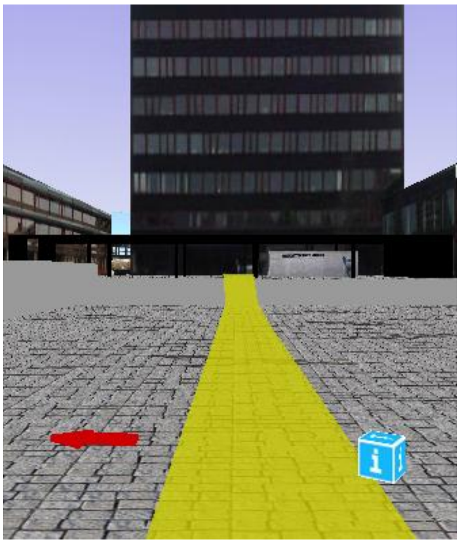

Figure 8.5: Screenshot from groundpath-mode. The translucent yellow sheen shows the path to the destination. The building ahead is “Niels Henrik Abels hus”.

possible, without becoming tedious. We wrote the code so that the viewpoint does not have a fixed starting point, but rather a default one. This way, the user can get a walkthrough that starts from any point on campus. The default starting position, obtained by selecting Free or Groundpath immediately prior to Walkthrough, is equal to the starting position of these viewpoints. Also, if Groundpath is selected prior to Walkthrough, the golden path remains visible throughout the animation. This is not actually an animation in the common sense of the word, because a user is still capable of interacting with the system - changing perspective or orientation. The path works like a magnet forcing the avatar to follow it and pulls the avatar towards the destination, although normal navigation is permitted on top.

Birdseye

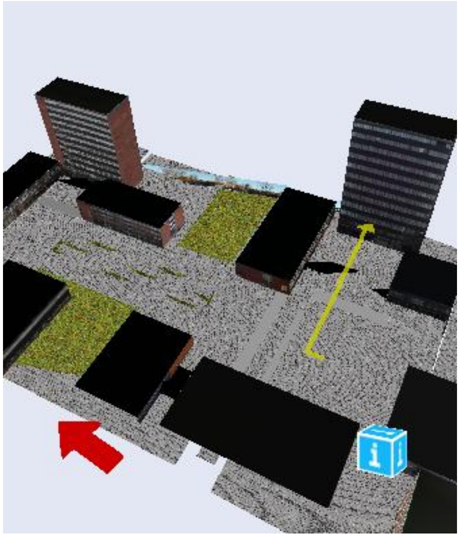

Figure 8.6: Screenshot from birdseye-mode in task0 (task0 is described in the next chapter). The yellow path points to the destination, in this case "Niels Henrik Abels hus".

The starting position of Birdseye changes between tasks, but is always located so far above the ground plane that it allows a clear view of most of the campus. It is usually centred on the starting position. Depending on the task, Birdseye also displays an arrow from the starting position to the target. When using Birdseye, gravity is turned off and the predefined movement type set to Examine.

8.6 Limited implementation

Developing a mobile 3D system for research purpose, we emphasised the navigation in 3D space at the cost of system functionality and robustness. A complete system would obviously require much more work. A network connection to download maps and contextual information, and a positioning system like GPS would probably be a requirement. Developing such a system in VRML only is not a realistic procedure. VRML and 3D data as a frame to display other information from a database or other systems, e.g. the Internet is more realistic.

Chapter 9

Experiment description

USER testing with real users is the most fundamental usability method and is in some sense irreplaceable, since it provides direct information about how people use computers and what their exact problems are with the concrete interface being tested. There are several methodological pitfalls in usability testing, and as in all kinds of testing one needs to pay attention to the issues of reliability and validity. Reliability is the question of whether one would get the same result if the test were to be repeated, and validity is the question of whether the result actually reflects the usability issues one wants to test (Koenen 1993, page 165).

9.1 Experiment method

In order to test our design, we decided to conduct an experiment that mimicked a scenario where the subjects acted out the roles of new students who used the system as an aid to perform certain tasks. The tasks where designed to simulate some of the chore a new student needs to perform on Blindern campus. In effect, the tasks consisted of identifying and moving to certain buildings on campus. We opted to use a qualitative method in the experiment for a number of reasons. Primarily, as explained in our methodology chapter, our research is explorative in nature, making a qualitative approach best suited. Also, using a few in-depth interviews instead of a widely distributed questionnaire was deemed less time-consuming and more in line with our requirements. This choice was also influenced by our available resources: We had only two researchers participating and only one available PDA. This meant that we had to set up a separate experiment for each subject. Clearly, a large-scale qualitative experiment was out of our reach. During the experiment, we encouraged the participants to "think aloud" while using the system. The Thinking Aloud protocol is a popular technique used during usability testing. During the course of a test, where the participant is performing a task as part of a user scenario, the participant is asked to vocalize his or her thoughts, feelings, and opinions while interacting with the product (Hom 1996). Thinking aloud may be the single most valuable usability engineering method. By verbalizing their thoughts, the test users enable us to understand how they view the computer system, and this again makes it easy to identify the users' major misconceptions. One gets a very direct understanding of what parts of the dialogue cause the most problems, because the thinking-aloud method shows how user interpret each individual interface item. The main disadvantage of the method is that it does not lend itself well to most types of performance measurement. On the contrary, its strength is the wealth of data it can collect from a fairly small number of users. Also, users' comments often contain vivid and explicit quotes that can be used to make the test report more reliable and memorable. The strength of the thinking-aloud method is to show what the users are doing and why they are doing it while they are doing it in order to avoid later rationalizations (Koenen 1993, page 195-196).

We chose to conduct our interviews in a less formal way, with the prepared questions used as an interview guide rather than a questionnaire. Many aspects of usability can best be studied by simply asking the users. This is especially true for issues relating to the users' subjective satisfaction and possible anxieties, which are hard to measure objectively. Questionnaires and interviews are also useful methods for studying how users use systems and what features they particularly like or dislike. From a usability perspective, questionnaires and interviews are indirect methods, since they do not study the user interface itself but only users' opinions about the user interface. One cannot always take user statements at face value. Data about people's actual behaviour should have precedence over people's claims of what they think they do. (Koenen 1993, page 209). Therefore, we should be aware that there may be a big difference between what users say (e.g. in questionnaires) and what they do (when using the system to solve a genuine problem). Users might say a system is good because they want to be polite, show appreciation for your efforts or encourage you to go on with your project. They might say a user interface is no good because it has no glossy graphic design but actually succeed with every single search because the user interface is clear and simple instead.

9.2 Scenario

We envisioned the scenario as follows: a first-year student arrives at Blindern campus. A major hurdle for new students is how to identify and accomplish the many tasks that must be accomplished before beginning their first semester: they have to register themselves as students, apply for and collect a student loan, familiarize themselves with the different buildings where lectures are held and so on. We picture a system that can help new student's accomplish these tasks by informing the student of the tasks that need to be accomplished and show him how to do so. This system would be implemented on a PDA or sufficiently powerful mobile phone. As the user walks around the campus, 3D data would be streamed to his device in near real-time, providing a concurrent 3D world on the device on which context-sensitive information could be displayed. We don't foresee such a system to become a reality for some years, even though the technology to create it exists today. Rather, sufficiently powerful mobile terminals to run real-time 3D graphics of relatively complexity must become much more common for it to be practical. Also, the system would require a high-speed wireless network, capable of speeds that surpass even today's GPRS-networks. A central part of this system would be a real-time 3D model of Blindern to serve as an interactive map. The system would be connected to existing campus systems such as Geographic Information System (GIS) to provide person-specific information. In effect, the system would keep track of what tasks the student has yet to perform. Unfortunately, creating a fully functional system such as this would be beyond the scope of this thesis. Instead, we found ways to simulate certain aspects of the system. There were also technical limitations in the VRML viewer that we had to compensate for.

9.3 Simulated parts

Because we wanted to prove that M3D is usable with current technology we believed it to be important that we outlined a system that relied on existing components. As explained in our design guidelines, we believe that M3D will be most useful as part of a larger system and not as a stand-alone technology. Therefore we applied the demand of using only currently available technology to the whole system. Though we chose to implement only the system components that used RT3D, we had to outline the rest of the system as well to verify this.

Wireless connection

We discussed several means of adding a wireless component to our system. In a full system this would probably be handled via a Bluetooth (Bluetooth.org 2002) or WLAN short-range connection, or possibly a mobile phone network via a protocol such as WAP. However, these methods were all deemed impractical for an experiment on such a small scale, so we decided to simulate a wireless network connection instead. Fortunately, there were only two types of data that needed to be downloaded: the geographical data and personalized information from the GIS system. These are considered individually below.

Downloading geographical data

As we could not stream geographical data to the PDA, we had to keep all the necessary information locally on the PDA. The only problem this represented was that, for reasons explained in the previous chapter, we had to store the system data as four separate files: One for each task. This meant that the user (or one of the researchers) had to close the current file at the end of each task and load another one.

Communicating with FS

We debated personalizing the tasks to each user, referring to them by name, etc, in order to give the impression of a personalized system. But we found that by keeping the tone informal, we could get much the same effect without hard coding user-specific information. In a complete system, data about each student would be downloaded from the university's central database, called FS. This database already interacts with several other systems. Creating an interface between our system and FS would be possible, but was deemed unnecessary to this prototype.

Cortona limitations

As described earlier, the Cortona software does not allow for a 2D GUI beyond the viewpoint selector and other VRML features included. Because of this we had to model the GUI as 3D objects. We could still make the GUI appear like a traditional 2D button interface, but we chose to use the possibilities offered by a completely 3D interface. We had planned to make some GUI elements like the compass 3-dimensional objects from the start, but we found that the other GUI elements could benefit from this as well. Firstly, modelling the objects in 3D made them appear as parts of the 3D world, relieving the user from having to "switch" between reading 3D and 2D elements in the same scene. We hope to make the presentation easer to read and reduce "clutter" by making the GUI and the 3D model appear as a seamless whole. This is in keeping with the augmented reality doctrine that the computer-provided information (in this case the GUI elements) should be seamlessly integrated with reality (in this case our 3D model). Secondly, having the GUI modelled as separate 3D objects allowed us to dispense with the "dashboard", and so minimize the area that the GUI occupies on screen. This is very important for us, since the PDA screen is so small that literally every pixel counts. Finally, the 3D design makes some GUI objects more usable in a 3D setting.

In particular, the 3D compass is more meaningful as a 3D object than as a 2D presentation since the user is capable of motion beyond the ground plane. To illustrate this point, consider the two cases: 1) The compass is shown in a traditional 2D fashion. There is an obvious discrepancy in that the compass' "north" needle points straight up, indicating that north lies straight ahead to the user. This works fine as long as the user is restricted to views parallel to the ground plane. However, when the user looks up or down relative to the ground, this presentation method becomes less obvious. 2) By giving the compass needle an extra dimension, the needle points due north irrespective of the user's orientation. In addition, this method makes the compass more intuitive while the user moves along the ground plane as it points in the exact direction The weakness of this method is that the compass object must be designed in such a way as to make its direction obvious when viewed from any angle. Our design suffered somewhat from the simplistic lighting scheme used, but it was still possible to discern the direction even while the arrow pointed straight away or straight towards the user (which is very rarely the case). We decided that the arrow outline was characteristic enough not to warrant a CPU-intensive lighting schematic.

Location and orientation detection

This is a very central feature in most orientation systems (Rakkolainen et al. 2000), (Allison et al. 2000). Using either GPS or cell identification, most mobile computer map systems are designed around location detection systems. Having access or not to such a system became a key to how the 3D model was designed. According to the EMBASSI papers (Kirste et al. 2001), the detail required of a map system is inversely proportional to the detail of the explicit location information. Without any such information provided a system will require a very high level of detail to ensure that the user recognized his surroundings. We designed our model to be as realistic as possible within the confines presented by the PDA. One of the primary things we were interested in was, of course, whether users could in fact see the correspondence between the 3D data on the handheld and the real world. GPS-like location detection mechanisms are not built into the commonly used PDA's today, and will probably not become commonplace for quite some time. But there are other ways of providing location data. For instance, the 3D data could be downloaded via a Bluetooth transmitter located near a visible landmark. Since Bluetooth has an effective range of less than ten meters, it could serve as a rough position indicator. The detail of the model downloaded would then have to be sufficiently detailed that the user could locate his own position more accurately. This was precisely the idea behind our scenario.

9.4 Blindern Campus

As can be seen from the accompanying map (figure 9.1), the Blindern campus consists of a large number of buildings located around several open spaces. The open area to the south is the Blindern square. The campus actually extends further to the south, with buildings that house the physics, chemistry and pharmaceutical faculties, among others. We chose to only model the buildings around the Blindern square and the area to the north, as we were limited by the amount of data that the handheld computer could manipulate at a time. This area also had the advantage of being recognized as the central part of the campus, and having a high density of different buildings and faculties. The area also contained several high-profile landmarks that could prove useful in navigation. The boundaries were selected so as to be formed mostly of buildings, to create the illusion of a wider space to the user. The southern boundary was made up of the buildings numbered 1, 2 and 15

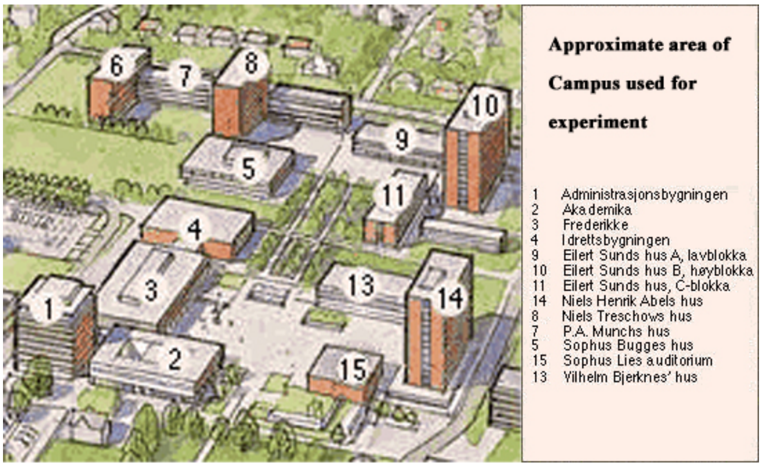

Figure 9.1: This map shows the approximate area modeled in the system and being used in the experiment.

in the map, as well as the university library. The library was built after the map above was made, but is located between buildings 2 and 5. The rest of the buildings formed a natural boundary for the area. The few remaining spaces between buildings were filled with photographs of the backgrounds to make a seamlessly closed environment.

9.5 The Tasks

As we wanted to test how the subjects used the system for different tasks we structured the experiment with this in mind. Each task the subjects were given represented something a first-time student was likely to do on their first day at campus. We decided to give the subjects a gradual introduction by giving them the easiest tasks first, and the more difficult ones later. What follows is a short description of each task. For a graphical presentation of the targets and suggested paths, see figure 9.2. This figure shows the routes of each task and the built-in paths used in the Walkthrough feature.

Task0

The first task was created as a test-task to allow users to become acquainted with the system before starting the actual tasks. The task included an example of all the different types of viewpoints as well as a practice target in the form of the mathematics building. This presented rather less of a challenge, as the building was in plain sight and decorated with a large sign bearing its name. Still, this task was designed mainly as a confidence-builder and to let the subject familiarize himself with the system, so we did not want the focus to be on locating the building

Figure 9.2: This aerial photograph shows the central campus. The recommended paths of each task are shown as yellow lines. All tasks started from the same spot, near the Air statue. This location is shown as a blue dot in the photograph. Source: (Finn karttjeneste 2002).

but on the interface itself.
Information Cube: Experiment and get to know the system.

Task1

This was the first "real" task, and therefore somewhat more difficult. The target was the administration building, where the subject was told to pick up his stipend. In a fully functional system, the user could be informed that he had received his stipend by the University's GIS system, which automatically tracks such information. In order to retrieve it, he must go to the administration building to sign a form. The task differed from the previous one primarily in that the target was not directly visible from the subject's starting point. The subjects had to round a corner in order to see their destination. We made a point of noting how the subjects reacted to this, which viewpoint they used and so on.
Information Cube: Go to the administration building to pick up your stipend.

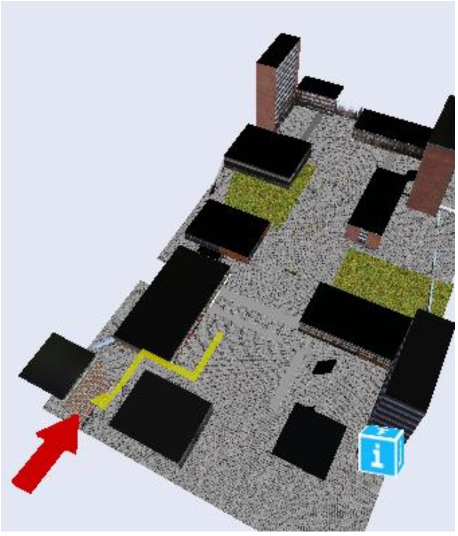

Figure 9.3: Screenshot from birdseye-mode in task1. The yellow path points to the destination.

Task2

For this part of the scenario, we postulated that user had received an oral message of where to go. We wished to test whether our 3D model was sufficiently realistic that objects could be recognized fro a brief, oral description. The subject was told to meet an acquaintance by the building next to the large green container. This landmark was chosen because it was unique, and large enough to serve as an obvious reference point. Using the container to identify the building seemed realistic enough. This task differed from the others in that it did not provide a Groundpath

or Walkthrough viewpoint, but rather six different Birdseye viewpoints. This was done to illustrate the fact that here the user was supposed to recognize a building from an oral or written description, and so had no formal aids to help him. The six Birdseye viewpoints were organized in three pairs. Each viewpoint in a pair showed the campus from a different angle, but were one displayed a wide area view, the second of each pair showed only the scaled-up central portion of its brother viewpoint. This created the illusion of "zooming" in on the campus area from three separate directions. The subject could only identify the target building using the "zoom" cameras and then had to use the wide area viewpoints to show the targets location relative to the users current position. This could be considered the most difficult task. The reason this task was not put at the end of the experiment was that the route for Task3 passed directly by the identifying feature: a large green container. If a subject had noticed this rather conspicuous item, they would have been able to locate it from memory rather than by using the system.
Information Cube: Meet me by the building behind the large green container.

Figure 9.4: Screenshot from birdseye-mode in task2. This birdseye view differs from the other tasks in that it does not show a yellow path marking the way but offers a “zoom” function instead.

Task3

The final task’s target, the main SV building, is located on the other side of the campus, making for the longest possible trek within the area that was modeled. It was otherwise almost identical to Task1, but while it was possible to see the administration building from the starting point (it is taller than the intervening buildings), the distance made the target building invisible to the user in this case. This remained true for almost all of the suggested path, except a short strip just above the

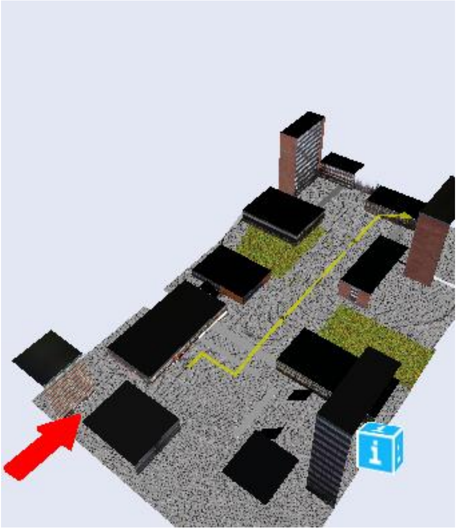

Figure 9.5: Screenshot from birdseye-mode in task3. The yellow path points to the destination.

stairs next to the fountain, where it was possible to glimpse the target between the other buildings. Information Cube: Go to the B-block of the SV-building.

9.6 Experiment organization and the role of the researchers

The experiment was organized as follows. There were two researchers available for most instalments of the experiment, designated A and B. To best simulate new students, each subject had little or no prior knowledge of the Blindern campus. Therefore we arranged to meet them at a location they were familiar with and then take them to the starting position on campus. Here, both researchers gave the subject a brief explanation of the system and how it could be used, allowing the subject to play around with Task0. The working language of the experiment was Norwegian. The subjects were instructed to communicate their thoughts about the system continuously to the researcher observing him at the moment, but other than purely technical difficulties, the researchers would provide no aid to accomplish the tasks. The short introduction to the system beforehand included:

- How to use the stylus
- How to use links
- The different forms of navigation in Cortona: fly, walk
- The different speeds: slow, fast, faster

- How to use the pre-programmed paths and views
- How the user interface worked and how the different parts worked

A textual representation of the introduction can be found in appendix B. The subjects will start their trek from an easily recognizable landmark represented in our model: the Air statue. This statue (although its actual name is not widely known) is a famous feature of Blindern's topography and easily visible. Here they will receive a tutorial from one of the staff and start up the system. The tutorial text (see appendix B) will be made available. The system starts up in an introductory mode, encouraging the subject to familiarizing himself with the controls. When the subject felt ready, researcher B left for the target of Task1 by a roundabout route and researcher A stayed to observe and take notes as the subject loaded Task1. Midway towards the target of Task1, researcher B took over observing the subject and researcher A headed towards the target of Task2. This switching allowed researcher B to observe how the subject acted when tackling a corner that blocked the target from view. When the subject reached the target of Task1, he was told that he did very well and was instructed to load Task2 while the researcher and subject walked back to the original starting position. During this time researcher B questioned the subject about his use of the system during the time the subject was

Figure 9.6: Illustration: Researcher Terje Torma instructing a model posing as a subject in front of the Air statue in december 2002. Note that the actual experiments were conducted in summer 2002.

too far away for his actions to be readily interpreted. During Task2 the subject was followed closely by researcher B until the task was accomplished. The subject and researcher then headed back to the starting point and loaded the final task. Upon completion of this, the subject and researcher were met by researcher A, and the researchers conducted an interview with the subject lasting 20-30 minutes.

Chapter 10

Experiment results

THIS chapter summarizes the most important experiment results that we obtained through observation, thinking aloud and interviews. The overall reactions we received were very positive. The subjects were all in agreement that the system was helpful and practical in accomplishing the tasks. Also, most of the subjects found the system fun and interesting to use and almost everyone commented on the exciting possibilities of the technology. Though not a significant result in itself, the fact that the subjects opinions were so positive is an indicator that the system could be functional outside a research setting.

Although the subjects had a varying degree of computer skills and little or no PDA- and 3D experience, we were impressed with their performance in learning how to use this new tool. They rapidly figured out how to use the stylus and the interaction means of the iPAQ and the 3D system.

10.1 Finds

In this section we have categorized the finds we see as most interesting in the light of our research questions. Finds from observation is combined with thinking aloud and interview inquiries.

10.1.1 Start/Stop paradigm

The subjects' most common use model consisted of intense use/no movement followed by moderate use/fast movement. We refer to this observation as *"the start/stop paradigm"*. We expected the use model to be of a more cyclical nature, where the user stopped to use the system, moved some way along the path, stopped to confirm his route and so on. Observing that most subjects were able to use the system at least to some extent while walking was very encouraging, as we had feared that the subjects would find the system too unfamiliar to have much attention "to spare". Although some users claimed to be able to complete tasks 1 and 3 using the system only while stationary, even these users were observed to use the system while walking. Some used it to check their route by glancing at the screen while in Birdseye or Walkthrough mode, while others were playing around with the system, examining the virtual campus from different angles. The fact that most users were able to use the system and remain mobile to such an extent is very encouraging for mobile 3D interface design.

The overall physical speed of the users suffered minimally from using the system. During three of the tasks, one observer walked alongside the subject. The subjects maintained an average walking speed while using the system. A few of the subjects halted and reoriented themselves along the way. This pause was timed to approximately 2-10 seconds and . The brief stop was, according to interview results, conducted either when a greater change of direction were in order, or the subject wanted to make sure he was on the right track. In addition to these rare “re-alignment” pauses, all but one subject stopped at the corner in Task 1. As described in chapter 9, we incorporated this corner to see how the subjects fared when they came to the end of their original line of sight (LoS). Virtually all the subjects reached the corner, stopped for about 5 seconds and loked around, then continued towards their goal. When asked why they stopped, the subjects either claimed to not have paused at all or replied that they were “playing around” with the PDA, and had to stop and re-align themselves when they came to the corner. This find contrasts somewhat with task 2 and 3, were the subjects also reached points were they went beyond their original LoS. But in these cases, only one subject made a similar pause as in Task 1. In these later cases, the boundary of the subjects’ LoS was less distinct, as it was marked by sets of stairs or trees, rather than a sharp corner. In these cases, the subjects seemed able to make the transition to the unfamiliar terrain without pausing in between.

10.1.2 Recognition

Most subjects were able to quickly make the correlation between the location and direction in the real world and the corresponding parameters in the system. This was especially obvious in Task0, when all but one of the subjects almost automatically turned to face north from the starting position, the direction displayed by the system. In this initial task all but one of the subjects pointed out the destination in a matter of seconds after either birdseye- or groundpath-mode was initiated (where the yellow path to the target building is visible).

The fact that some subjects were capable of using the system to get a detailed impression of were they were supposed to go and then moving straight to their target with minimal double-checking speaks positively for the model’s realism. Especially since the users were able to do this even when the target was out of sight from their original position or there were a lot of other people present. One must keep in mind that the routes were relatively easy, in that they were relatively short (the longest being slightly less than 200 meters from start to finish) and that there were no vehicle roads present. The area is still very dense and there are a lot of trees and smaller structures that obscure vision. Overall the area could be said to be slightly easier to navigate than a similarly sized area of inner city.

The level of detail appeared to be sufficient. The use model adopted by most subjects, heavy use of the system prior to movement followed by light use and walking a straight path to the target, indicates that our attempt at near-photorealism has been effective in making recognition nearly instantaneous. Several users commented favorably on this as well.

Subjects used different 3D characteristics in orienting themselves and locating their destination. The size and colour (including materials: bricks, concrete, wood) of buildings as well as their relation with nearby objects were the most prominent.

10.1.3 Mental map

The subjects were all capable of using the system to get a detailed impression of where they were supposed to go and then moving straight to the target. They did this even when the target was obscured by buildings or otherwise invisible. Most of the subjects were confident that by using one walkthrough or birdseye view mode they had the route planned mentally. But as mentioned above, most of them checked their progress while underway.

A few of the subjects, after localizing the target in the system, commented on why they should bother go walk the path to the destination since they already felt they had been there. This implies that not only was the model realistic, but it must have enabled the subject to construct a mental map of an area they never had visited. The degree of detail of this mental map may be expressed by this quote by one of the subjects:

> *“Do I have to go all the way over there? I already know what it looks like.”*
>
> -Research Subject

10.1.4 Interaction

Automating the process of navigation worked very well. The users were still capable of orienting themselves without being totally in control of the movement in the 3D world. We put considerable effort into making the camera movement smooth, but the results were not entirely satisfactory. However, the experiment showed the importance of using smooth camera movements to allow users to easily understand the path. Rapid shifting of camera perspective in walkthrough mode in some areas in the model could cause confusion and orientation difficulties, as some subjects commented.

> *"This rapid shift of viewpoint could be confusing. Of course, I understand were I am supposed to go."*
>
> -Research Subject

10.1.5 Interaction

Using clickable symbols for handling the GUI-interaction worked well. The infocube symbol presented as a rotating blue cube with the commonly familiar "I"-letter was adequate for the subjects to understand how they should start using the system and get help. One feature we had implemented that was not observed in use was the possibility to navigate at the same time as reading the board when infocube was enabled (because of transparency). This may have been because the small amount of information displayed on the board was easy to remember.

The pull-up menu system in Pocket Cortona caused a bit of frustration at first. The many options and the small letters and button-area caused some subjects to make mistakes. But after a few minutes almost all the subjects managed to change avatar speed and viewpoints rapidly.

Using the stylus and click/drag-mechanisms to interact with the system worked better than expected, but many users commented that the rendering speed in the model was too slow. It was also observed that few of the subjects used examine-mode to look up/down in the 3D world. This feature was not required to accomplish the tasks, so there was little incentive to use it.

Most of the subjects expressed a preference for the Walkthrough viewpoint, and most of these used it in conjunction with the Groundpath option. This combination showed the golden path, and then had the camera follow it automatically to the target. The subjects reacted very positively to the system's ability to simulate motion in 3D. Many subjects held this to be the most striking improvement upon traditional maps. To us, the designers, 3D movement is quite natural and we considered this to be a small bonus to the system. But to the subjects, seeing a 3D runthrough of the path they were walking was a new and exiting experience. Some of this excitement was undoubtedly caused by the novelty of system, but many subjects relied solely on moving views to perform some of the tasks. This indicates that subjects were able to utilize the information presented in this way.

Many of the subjects found it difficult to use the "Free" viewpoint. They mostly used this viewpoint during Task 0 and experimenting with the system. The most confident subjects used this viewpoint more, even while walking. One subject used the “Free” viewpoint almost exclusively to locate the targets of the different tasks.

When asked about the interface during the interview, most users had little to say about the interface organization and layout. But they inquired about including some features such as a search option in a complete system.

10.1.6 Technical issues

Technical limitations were of greater importance than we expected. We originally surmised that the difficulties in getting the system accepted by the public would be non-technical in nature. The test-subjects readily accepted the scope and premise of the system, but almost all of them commented that some technical limitations had to be overcome. The most striking evidence came from the questionnaire, were the subjects were asked to rate certain aspects of the system. The subjects’ answers were very similar in that they held the screen size and other aspects inherent of the handheld platform to be sufficient, but rated aspects such as speed and responsiveness lower. These technical limitations were mostly due to the VRML viewer we used. A complete system could probably be designed using only VRML, but this would require a fully functional mobile viewer. The Pocket Cortona showed poor handling text nodes and interoperability with other applications, such as a browser.

We expected the users to have problems with the pen interface of the PDA. We predicted that this difficulty would result in little use of the “Free” viewpoint, which only provided this mode of navigation in the 3D world. Our results were slightly different Most users expressed difficulty in dragging the pen on the touchscreen an a preferance for the automated movement views in the interviews. But during observations and while thinking aloud at least two users displayed great skill in using this mode of navigation.

10.1.7 Other finds

Even though the trees in the model were not truly 3D objects and the texture used was a “cartoony” representation, one subject commented on their inclusion:

> *I see you’ve even got those trees into the model.*

- Research Subject
The subjects commented on the use of navigation "stages". A valuable insight we received was the concept introduced by subject number two. He commented that organizing the paths in separate stages, were each stage corresponded to the limits placed on visibility by the area, would allow for quick and easy navigation. When we designed the system, we had considered the idea of stages, but thought that

Blindern campus was small enough to be considered a single "stage". Basing the stages on the user's field of view would fit very well with the use pattern the subjects employed: when they reached the limit of the area they could see from their starting position, they invariably slowed their pace or stopped briefly, checked the system, and continued. However, it could also be argued that using longer stages would allow the user a kind of X-ray vision by letting him see what lies beyond his immediate field of vision. Finding a way of balancing these two concerns could be an interesting subject in itself. For the purposes of our system we would probably allow the subject both options.

When the subjects were asked to think of new scenarios such a 3D map system could be used in, we got some indication that the 3D technology in itself should act as interface for a larger system. An airport guide was one idea, and new features included in a campus guide was also mentioned. In both cases the functionality for problem solving was particular prominent. The airport scenario mentioned by a subject was an idea highly appreciated by the subject. He was constantly on the move and visited airports frequently. A 3D map system for displaying easy and fast routes to gates and facilities, along with flight information could have been of considerate value to him he said.

10.2 Sources of error

Despite our best efforts, we were unable to procure a representative cross-section of all age groups and levels of technical competence who would be elligible to use the complete system. Most obviously, our subjects were from a narrow age group, with five subjects being 26-27 years old and one being 23 years old. During the observation we were unable to take advantage of video/voice recordings but instead we draw positions on paper and took notes.

The experiments were for the most part carried out according to the original plan. For a full description, see the previous chapter. However, we found that the initial version of Task 2 was poorly designed, and so this task was modified after the first three experiments. The original Task 2 had the subjects identify a building in the model from an oral description, ie: "Locate the building next to a large shrubbery". Unfortunately, we found that locating this object in the model was next to impossible, as its small size and inconspicuous colour made it next to invisible from the Birdseye viewpoint. In hindsight, the shrubbery was not a very realistic waypoint either, as it was obscured by the sign that showed the name of the building. For the following four experiments, we redesigned the task to read: "Locate the building behind the green container." The container contrasted more markedly with its surroundings, and was a unique feature that was easy to describe. In addition we provided the user with three distinct Birdseye viewpoints and the ability to "zoom" in a secondary viewpoint. The primary viewpoint showed an area large enough to be easily recognized, but the container was not immediately apparent. Switching to the secondary view showed a much smaller area, but the container was obvious. The user had to identify the general area from the primary views, and then pinpoint the exact location using the secondary viewpoints.

While the subjects were unable to complete the original Task 2, the revised version was better received. The subjects quickly matched the verbal description the correct location, but the limited number of subjects mean that we can not draw any meaningful conclusions on whether the 3D model was compatible with verbal way-descriptions.

Chapter 11

Discussion

In this chapter we compare the results described in the previous chapter with our theoretical basis. From this analysis we hope to gain some insights into the potential of mobile 3D systems and how users cope with the shared attention difficulty. As described in earlier chapters, we draw on theories from mobile IT, real-time 3D, augmented reality and cognitive psychology to systemize the data from our experiment.

Recall the main research questions from chapter 1:

- **How can a mobile 3D system be designed to compensate for shared attention?**
- **Can users utilise our system?**

As explained in chapter 1 we had to limit ourselves to a pre-study in order to answer the first research question. This pre-study resulted in the system requirements listed in chapter 7. The system described in chapter 8 serves as an example system that aims to fulfil this system requirements. In order to answer the second research question we devised a practical experiment as described in chapter 9. In this chapter we discuss the results from this experiment and how they can shed light on our second research question.

11.1 Interaction

As pointed out in the interaction design chapter, many of the most commonly used design guidelines for user interfaces were of little help during our design phase. Several authors have pointed out that interface design guidelines like those of

Nielsen (Koenen 1993) need to be modified when applied to systems incorporating special technologies like RT3D (Köykkä et al. 1999).

This is perhaps to be expected when designing a system that is somewhat unusual: It is very difficult to create guidelines that cater to every possible system without becoming so general as to be obvious. Also, some guidelines derived from specific mobile systems projects, like Pascoe et. al.'s fieldworker study (Pascoe & Ryan 1999) were too specific to be applicable.

Further, the Cortona VRML application we used to render and present the 3D world put severe restrictions for how the design can be implemented. Creating a system that incorporated the features we needed to compensate for shared attention required much work and some pragmatic design solutions. This made applying the guidelines of other researchers even more difficult, as their admonitions were sometimes impossible to implement in VRML.
Based on this experience we think that when designing graphical intensive applications for mobile IT use, developers need more clearly defined rules - especially regarding shared attention. Attempts at creating guidelines and theoretical frameworks like Pascoe et. al. (Pascoe & Ryan 1999) and Beck et. al. (Beck et al. 2002) are to be commended, but there is still a need for further work on these subjects.

Manually navigating (using the "Free" viewpoint) the 3D world was perceived as confusing by the subjects (see find T3). Fortunately we had taken steps to deal with this issue and provided the subjects with several means of partially-automated navigation, orientation and movement functions (see chapter 7). It became clear that most subjects favoured these functions rather than navigating manually with the basic functionality that was included in the VRML browser. The difficulties they faced in using the basic means of navigation and the enthusiasm they displayed for the easier forms of navigation led us to conclude that *Quick and easy navigation is of paramount importance in truly mobile use.* Using partial automation and predefined viewpoints, as we did, is perhaps not the best solution, but we were forced to rely on the hardware options available on the iPAQ.

The main technical limitation we observed was the hardware tools for navigating - the input mechanisms. The pen and touchscreen interaction tools along with "touch and drag" to control speed and direction was deemed impractical or nearly so for 3D navigation. This last issue might be caused by the lack of experience with 3D navigation in general and unfamiliarity with Compaq iPAQ use by the subjects. The subjects that preferred the "Free" viewpoint generally had experience in using PDA's. The subjects experienced difficulty in switching between navigating in normal mode and examination mode (see section 8.5).A hybrid form or a quick change between the two movement types could have been helpful. Switching through a menu popup option in the Cortona VRML browser was perhaps too difficult or "fidgety" (the popup menu was very small, see figure 8.2) for practical use. The result of this is that the movement of the avatar and changing angle or direction of perspective are two distinct navigation modes. This is incompatible with how we normally navigate in everyday life (example: making the avatar look up while at the same time walking is not possible using the Pocket Cortona browser). On a desktop computer this problem is solved using two navigation input mechanisms (mouse and keyboard) rather than one on the Compaq iPAQ (the pen). The keyboard sets the avatar in motion whereas the mouse controls the perspective angle. However in a mobile context this navigation problem can not be solved by using two input mechanisms due to the fact that both hands are already occupied; one to hold the device and one to operate it. *There is a need for a more intuitive and simple means of navigating 3D worlds while mobile.* It could be interesting to see how the input devices in newer PDA/mobile phones could be used in 3D navigation. Finding a better input technique and combining it with navigational "aids" built into the interface could be the optimal solution.

The experiment subjects used navigation in different ways. Some used a combination of several navigation aids whereas others used one single aid throughout the experiment. Though using a mobile 3D system was a new experience for all of the subjects, the group could be divided into confident and unsure subjects according to how they utilised the system. Subjects that showed a particular interest in mobile 3D seemed to use different aids during each of the tasks. The more timid subjects found one favourite aid and stuck to it during all the tasks. The confident group had in general more computer experience and showed more interest than the nonconfident group. This subject had no prior experience with 3D navigation at all. One could expect that the more confident subjects would be more likely to use the navigation modes that gave the most control. This was, however, not the case. The more confident subjects almost always opted for the "Walkthrough" or "Birdseye" viewpoint. The less confident also favoured these viewpoints, but spent more time with "Free" and "Groundpath" at first. These trends were noticeable, but not absolute, as there were exceptions. In particular, the experience factor did not apply to the subject that made the most expert use of the "Free" viewpoint. The wide variety in methods to locate their targets and orient themselves pointed to that *different navigation methods appeal to different users.* Offering a variety of navigation methods did not seem to confuse the subjects, but rather encouraged the subjects to experiment and find a navigation method they preferred. The differences between confident and unsure users suggest that having several use modes may improve users' performance at higher skill levels.

11.2 Recognition and realism

One of our most encouraging finds was the fact that almost all the subjects grasped the connection between the 3D model and the real world almost immediately. Firstly this was shown by the way they automatically aligned themselves with a given viewpoint on the PDA, and secondly by the fact that they were capable of recognising buildings from the 3D model very quickly. This was evidenced by the remark:

> *"Do I have to go all the way over there? I know what it looks like."*
>
> -Research Subject

During our design phase we focused on realism almost exclusively to obtain recognition, and our results seem to indicate that this paid off. But this reliance on "realism" holds a potential pitfall: In what degree is "realism" possible with RT3D on a PDA? With the PDA's small screen and limited resolution this seems a difficult task. In addition, the technical limitations that the PDA poses results in a lower framerate and much less detail than would have been possible in a RT3D model on a stationary computer. And even if fully developed for a stationary computer with a state-of-the-art 3D engine, true realism would simply not be feasible.

However, when asked, our subjects compared our system's reliance on 3D favourably to traditional maps. Based on our results it appeared that our modelling approach aided in the subjects' "mapping" of the navigation data to the real world. As we have seen, the results from the article "for maps and guidelines" (Brown & Perry 2001) presented in section 5.1 emphasise the importance of "mapping", the process of connecting the physical world with social information. Even though our 3D model was nowhere near realistic, it did mimic many effects that the subjects could recognise from the real world. These effects, including perspective, vivid colours and familiar shapes and outlines, were apparently realistic enough to convey a sense of familiarity, as evidenced by the subjects ability to recognise buildings and features in the real world from the 3D model. Perhaps it would be more correct to refer to these features as providing detail rather than realism.

Even though it seemed evident that the subjects were to a high degree able to "map" between features in the real and 3D world; what about the use of "mapping" to convey social information? As explained in section 5.1, traditional maps have trouble conveying the social aspects of scenery features. Could these "realistic features" aid in communicating such information? Unfortunately, we did not design our system to do so, since we were primarily concerned with the shared attention dilemma. But the most obvious way of providing such information would be through annotations. This could not compete with having a local guide pointing out places of interest and their significance. Trying to prompt the user with pushy annotations could make use of the system irritating.

But there is perhaps another way to provide social information through a 3D map. The more detailed the representation of a building or object, the higher the chance of capturing a feature that a user finds interesting and wants to know more about. An interesting point in this regard was that one of the subjects completely ignored the statue he stood under (the Air statue) until he saw it in the model. This piqued his interest enough to ask what it was supposed to represent, who made it and so on.

To summarise, although our insistence on realism seemed to have paid off in rapid recognition, there is a very real possibility we ended up just provide a more detailed alternative to a traditional map. This may hold significance for a future system, where one could perhaps focus on emphasising detail rather than realism.

Another point of consideration is the inclusion of landmarks. Introduced in chapter 5 landmarks act as points of reference in the real world and 3D models alike. Deciding on what landmarks and features to include was a difficult task. We did not know what landmarks or features people would prefer to use to orient themselves. We therefore included as many as possible, and when it came to a choice, we selected the prospective landmark that translated best to a 3D model. The number and quality of the landmarks included in the model is intimately connected to the issue of what level of detail is required in the model. More detail means a greater chance that the subject will recognize a feature, but it also results in a greater chance of confusing the user. In keeping with our goal of realism and the theories of (Butz et al. 2001), we chose to include as much detail as possible without suffering lowered frame rate.

The subjects were given a reference point at the start of the experiment (all tasks started next to the Air-statue). This can be considered "cheating". Had they been forced to navigate to the starting point themselves, they might have taken longer to recognize their surroundings. As explained in chapter 9 the reasoning behind providing a fixed starting point was to simulate an actual use-setting. If the system had been completed according to the full design, the PDA would have had to download the 3D model data from a short-range network transmitter.
The location of the transmitter that we simulated (in the base of the "Air" sculpture) is perfect for our purposes, but very unlikely in a real setting. A more realistic place to put a transmitter would be inside a building. This would allow a PDA that came near the building to establish a connection and download the data. But in this case, the initial location of the user would then be much more uncertain. If for instance, the transmitter had been located inside the closest building on the campus, the Fredrikke building, the user would only have known that he was within a few meters of the building. He could be located on the other side of the fountain or even outside of the 3D model altogether, since the Fredrikke building forms part of the boundary of the modelled area to the west.

> *This is almost too easy: With this, one simply cannot go wrong.*
>
> - Research subject

The fact that the subjects found the tasks to be easy to accomplish could be caused by just that: That the tasks were too easy. During the design phase we were con-

cerned that we might make the tasks too difficult. The first rule of thumb about designing for shared attention is to stress simplicity, after all. But we did incorporate some challenges into the tasks.
In retrospect, the tasks could have been designed to be more demanding. Doing so would have allowed us to draw stronger conclusions. But during the design phase our chief concern was too determine the validity of using a real-time 3D system while being mobile. Making the navigation tasks too challenging could have resulted in us testing the subjects' navigation skills, rather than the system's ease of use. Another factor we had to consider was the second cognitive factor of shared attention: practice. We did not have the opportunity to determine how practice would affect users' performance, but we found it reasonable to assume that it could improve with time. To compensate for this lack of user practice we wanted to make the system easy to use from the start. Of course, these two things are not the same, and it may be that subjects would have reacted differently to a more difficult system, even if they had hours of practice with it.

11.3 Realism vs. symbolism

During our design phase we found that we had to make concessions to our goal of realism because of the constraints posed by the PDA platform. The advantages of stylised representations, greater clarity and less taxing for the PDA, were so great that we found we had to include such objects in our model to obtain a functional system. As described in chapter 8, this resulted in simplification of buildings and features to make them easier to render. During this process we found that we had to convey the impression of an object without complete realism. This is perhaps best illustrated by the way the stairs and trees were modelled. Both were simplified to the point of being stylized. This can be seen as a step towards a *symbolic* representation of the world, as opposed to the *realistic* representation we aimed for. A symbolic representation is closely related to the concept of metaphors, described in section 5.1. However, as (Preece et al. 2002) points out, a metaphor is more than a simplified picture of something. The goal of a metaphor is to create a mental model of expectations in the user rather than just act as a label. During the system design we saw simplifying the structure of objects as a necessary compromise caused by the limitations of the iPAQ, and different from using metaphors. But as the project progressed we found that there were many similarities between symbols and metaphors. Both can be seen as *representations* rather than realistic *simulations* of objects. They usually take the form of stylised drawings of a real object and their use takes it for granted that they are recognised as what they represent.

While we avoided the use of metaphors in the 3D world, we did utilise them in forming the user interface objects. An interesting finding that illustrates the potential pitfalls of metaphors was our 3D compass. The compass was designed as a red arrow, similar to the needle pointing to the north on compasses the authors were familiar with. As explained in chapter 8, this design was chosen because it was simple, had a clearly defined 3D shape, and we thought was easily recognisable. It turned out that only two of our research subjects had any experience with using a compass, and so had the mental connection that "red arrow points northwards". The rest of the subjects needed the metaphor explained to them, though most grasped the use of the arrow as a navigational aid intuitively. It is also interesting to note that the two subjects who grasped the compass metaphor also had the most experience in using normal maps, and so understood the entire system as a "map-metaphor". They compared the system to using a normal map when interviewed and used the terminology of normal maps when explaining their use of the system. Clearly, the use of metaphors can cause confusion, or even create unintentional connotations to users.

It was precisely these dangers that made us avoid metaphors and symbols as far as possible in designing our 3D world. But the advantages mentioned above make more extended use of simplified representations very interesting for M3D. One could imagine that if the goal of realism was relinquished, one could create an extremely efficient M3D system. The simplified architecture would mean smoother navigation and movement, and more processing power for running more complex scripts. The greater clarity afforded by stylistic geometry and textures could in theory make the system more intuitive and require less attention. If this idea was taken to its logical conclusion, such a 3D model could have been composed entirely of billboards. In this case, all objects in the 3D world would be 2D "plates" with textures composed of symbols and maybe even texts. Or, to use terminology from Augmented reality: The features would be replaced by their annotations. Such a model could still communicate the spatial relationships between objects, their relative size and even outlines, and so retain the advantages of RT3D.

In our own experiment, the results indicated that simplifying object representations did not make recognition more difficult. During the experiment, the subjects expressed no difficulty in recognising the stylised objects, and in fact some subjects commented on their inclusion as something positive.

> *I see you’ve even got those trees into the model.*
>
> - Research subject

This ease of recognition could have been because the stylised objects were relatively few and interspaced with more realistic ones. If the subjects recognised the realistic buildings and features first, they could then infer what the symbolic features were supposed to represent from their relative positions. But the speed and accuracy with which the subjects identified these objects seem to indicate that users were able to identify them on their own merit.

The downsides to using metaphors can also be applied to symbols. As described in chapter 5, they can be culturally or logically misinterpreted, they may be too constraining, conflict with design principles, be poorly designed, the lack of detail, or limit the designer by a certain metaphoric paradigm. In addition, symbolic representations loose many of the details that distinguish it. In creating a stylised representation that emphasises clarity, details that could be important will be lost. As evidenced by the fact that users reported using a host of different cues to identify buildings and features (colours, material, size, and so on), these details can be very helpful. Realistic and symbolic representations apparently both have their strengths and weaknesses. A fully symbolic representation of a real world area may be as unfeasible as a completely realistic one, but using a stylised approach may make sense even in cases were performance is not the issue. So when should one strive for realism and when is a more symbolic approach preferable? The similarity between symbolic or stylised representations and metaphors may offer some insights.

There is an interesting parallel between the metaphors described by (Preece et al. 2002) and Vinson's (Vinson 1999) guidelines for *landmarks*. Both should be clearly understandable and distinctive from other objects to avoid confusion (Köykkä et al. 1999). The stylised nature of symbolic representations often make them stand out when put next to more realistic objects, making them more distinct. As described in chapter 8, we utilized this effect when designing the user interface objects, making them stand out from the 3D campus model. Additionally, both metaphors and landmarks are usually representations of concrete, everyday items that the users have intimate knowledge of and experience with. The idea that well-known objects are better suited to simplified representation was corroborated by our own experiment results. In our experiment, we found that the simplified objects that were most easily accepted by the subjects were those they had the most intimate experience with: The stairs and trees were immediately recognized, while the garbage container that was the target of “Task 2” was more difficult to recognize.

The similarities between metaphors and landmarks seem to indicate that simplified representations are suitable to act as landmarks in a 3D world. However, there is a drawback to simplifying the structure of landmark objects: According to Vinson (Vinson 1999), abstract models are less suitable as landmarks because they are more difficult to remember. It is also possible that symbolic representations are perceived as being more general, and so are not unique. There is no point in using a tree as a landmark if there are a hundred identical-looking trees around. However, Vinson's guideline applies to non-figurative art rather than stylistic representations. It is possible that stylistic representations of everyday objects may mitigate the problem somewhat.

Also, the circumstances of their use may make up for the lack of individuality of symbols: Symbolic representations can be made unique, for instance by varying textures and colours. But this counters the advantage of optimization somewhat, and so is probably not ideal. By presenting them as groups, symbolic representations become more "memorable". While a single tree lacks distinguishing features, three rows of carefully planted trees, such as was present in our model, makes for an easily-recognised landmark. The advantages of optimized geometry of stylized representations are also compounded when representing groups of similar objects. Another way of increasing the distinctiveness of symbolic representations is placement. By placing them in conspicuous locations, symbols may become more distinctive. An example from our model becomes that Niels Henrik Abel's building, has a set of stairs in front of it. The stairs themselves are similar to the others in the model, but the combination of stairs *in front of* a building is distinctive. Finally, even though stylized representations are similar to others of its kind, they can be *locally* distinctive. If only one side of a square has a fence, this can help orientation, even though there are many other fences in other locations of the 3D world.

11.4 Shared Attention

In the experiment phase we observed that the subjects managed to operate the M3D system on the iPAQ, converse with us during the "thinking aloud" inquiry method, and reach their goal of destination within a limited timeframe. This suggests that they managed to perform the three activities at least approximately simultaneously. Now it remains to discuss to what extent these tasks where carried out "simultaneously". In this section we present two different ways of looking at our results.

Statement 1: *The subjects did not need to use the system while walking and conversing because they quickly constructed a mental map making the system superfluous while they walked towards their destination.*
If so, our observation of the subjects using the system while walking and conversing could be misunderstood - they were simply checking the system against their mental model or fumbling with the iPAQ for other purposes, and thus giving the system a minimum of attention. If this was the case our GUI design effort had little to do with directing the subjects attention other than helping to obtain the proper mental model prior to walking. Instead of helping the subjects coping with shared attention, our system let them build a mental model to complete their tasks. Note that the subjects must have memorized the modelled area very quickly, however. They had only the first "trial-task" (Task 0), followed by maximum 30 seconds at the start of each subsequent task before they started walking. Also, that the subjects primarily relied on their mental model as a map does not necessarily make our system superfluous: At several points in the experiment, the subjects checked their progress against the system. They used Birdseye-, Walkthrough- or even the Free- viewpoint to compare the model with their surroundings while remaining in motion.

Nevertheless, if this is the case, we can say we have succeeded in creating a realistic 3D model that is useful in a mobile setting. However, this would mean that the system was not directly successful in investigating shared attention.

Statement 2: *The subjects used the system continuously as a reference while at the same time walking and conversing.*
As noted in 1) we observed the subjects interacting with the system while walking and conversing. If the users did not build a mental map of the area before walking towards their destination, the use of the system must have served as their primary reference. In this case, use of the system was more in keeping with our expectations. Several finds substantiates this view: Several subjects were observed to update the system's view according to their movements in the real world. Also, virtually all subjects stopped when they reached a point that they could not see from their starting position in task 1 (even though they claimed not to during the interviews). One could argue that "toying" with the system is the same as *using the system*. To what extent they used the system can be considered irrelevant as long as there are some indications that the subjects paid some attention to it. The fact that they paid some attention to the iPAQ and the system hopefully states that the system gave some meaning. We are not concerned about what this meaning could be. We observed that the subjects shifted eye-focus rapidly between the device and the surroundings. If they merely looked at the system for fun or because they were bored is not of our concern because they did it at the same time they were walking and conversing.

The two statements are distinctly different because how the subjects used the system can be interpreted in two ways. In the first statement, the subjects' interactions with the system is considered less crucial to completing the task. In the second statement the subjects' interactions are all assumed to be task-related. In this interpretation the subjects were *dependent* on using the system while moving.

11.4.1 Comparison of statements

Perhaps both of these views are too extreme and the truth lies somewhere in-between. We can be reasonably sure that the subjects did in fact create a mental model of the campus and used this as a reference while performing their tasks. However, the results also point to that at least in some cases, they relied on comparing the system to their surroundings. The most obvious example of this was "'ref til snu-operasjon"'. During system design, we did not anticipate the importance of a mental model, at least we did not foresee that the system could be primarily useful as to create such a mental model. It is possible that the reason the system was so effective in letting the subjects create a mental model of the area was due to local factors or our design choices. The local factors include the limited area modelled and the relative complexity of the campus topography. The area modelled in the system is relatively small, just 70000 square meters (0.07 square kilometres).If the area had been larger and thus contained more buildings and features, it would have been more difficult to commit to memory. As explained in the System Description chapter, because of hardware constraints of the iPAQ, the area modelled had to be carefully balanced against the level of detail we could achieve. Also, the Blindern campus is relatively clearly set out, especially if the subject relied on the Birdseye viewpoint. If we had chosen to model a more densely built area, like downtown, buildings would have blocked the subject's views both in the real world and in the model. If the area had been more complex, as in more and taller buildings, it would have been more difficult for the subject to create a mental model. The subject probably would have had to spend more time committing the 3D model to memory and would perhaps have been forced to rely more on the 3D system itself.

There were also design decisions that influenced the ease with which the subjects committed the 3D model to memory: Our efforts to make the model more easily understandable made it easier to remember as well. By simplifying complex objects and exaggerating traits that made features unique we created a model that was optimized for memorization (chapter 7).

11.5 Revision of our shared attention assumptions

Another issue that is highly relevant to how our experiment results should be interpreted also concerns how we understand the concept of shared attention. During the design phase of this project, we focused our efforts on making the M3D system demand as little attention as possible. This approach assumes that dual- tasking as described in chapter 6 is possible and even desirable for users of M3D systems. However, our results indicate that this may not be the ideal approach. While we did observe the subjects moving and using the system at the same time, as mentioned above this use could have been confined to simple confirmations. Also, the subjects naturally adopted a "stop/start" paradigm that consisted of heavy use/no movement and light use/fast movement and could alternate between the two several times during a single task. This last effect was particularly noticeable in Task 1, where all the subjects stopped to re-orient themselves when they encountered a corner.

This "stop/start" paradigm was very similar to the two use-modes described by Pascoe et. al. (Pascoe & Ryan 1999). Although Pascoe et. al.'s use-modes were designed for field-workers and not intended for rapid switching during tasks, the parallel is obvious. Recall from chapter 6 that the fieldworker application had two user-interfaces: one for high-attention functionalities and one for low-attention ones. Each of these relied on different modalities. This arrangement would probably not be suited to our M3D system, as the subjects changed use-mode very rapidly and were able to rely on visual output even while moving. We did offer navigation functionalities that provided users with several different balances of control versus attention-demand. But as explained in chapter 5, these were intended to find how much attention the users could apply to the system. As described in chapter 6, we also foresaw that users would possibly adopt a dynamic use-pattern where the rate of movement varied according to how much attention the user applied to manoeuvring.

These finds can be read to indicate that our shared attention design was too focused on allowing dual- tasking. Part of this problem lay in our level of abstraction in data collection. As described in chapter 6, we would allow the subject to stop for a few seconds to check his progress and then keep moving, while still considering him to use shared attention (albeit multi- tasking rather than dual-tasking). This distinction was chosen to free us from an in-depth analysis of the subject's mental processes. The subjects' behaviour indicated that this level of abstraction was too high. Given that most users adopted the stop/start paradigm, the division between dual- and multi-tasking could have been made more fine-grained without necessitating a detailed discussion of shared mental resources.

In retrospect, our premise of designing for dual-tasking may not have been the most appropriate. Instead, designing for a dynamic use mode may be a better way of handling the shared attention problem.

One way of doing this would be dividing the features into two groups: one requiring a high level of attention and the other requiring a very low level. In the first group would go navigation modes such as "Free" and "Groundpath", while the second group would consist of viewpoints like "Walkthrough" and "Birdseye". Display and input means could also be optimized for the different use-modes. It would be important that switching between the two modes could be done quickly and easily to allow rapid changes in use- mode.

In light of this, our initial suppositions may need revising. Downplaying designing for shared attention in favour of rapidly-changing use-modes combined with an increased reliance on mental maps might work better than minimizing the attention required by an M3D system.

Identifying issues like these and revising one's assumptions are part of the scientific process, and can be a valuable contribution for later researchers. We hope that these finds can be of value to others who are interested in designing M3D systems.

Also, the techniques we developed for dealing with shared attention can still be useful in an M3D system and may contribute in further research. Though automating navigation, dividing tasks into visible stages and other features may not be directly helpful in coping with shared attention, they can still have a positive effect. Many of these design features reduce the workload of the user, making the system more usable, and the experiment showed which of these features the subjects preferred.

11.6 A special case?

During our experiments we were impressed with the speed at which the subjects learned to use the program and how easily they made use of the system. But one subject, subject 6, went against the grain in almost every way. She spent a lot more time on each task, made several false starts and in almost every way failed to make use of the system.

Here is a list of observations with subject nr.6: She...

- did not reorient herself at the statue (task0).
- used more time.
- did not switch between system and real world often.
- asked many questions.
- spent more time looking at the system than at the real world.
- criticized the experiment setup and the researchers handling, saying that it would have been better to conduct the interview while the tasks were carried out and that the researchers should have been more forthcoming with helping her use the system.

Although her performance at the tasks was mostly below that of other subjects, she did provide some valuable insights as well. For instance, she claimed to have made use of the colours of buildings to recognize them.

This consistent failure and the results of the interview seemed to suggest that the subject simply failed to grasp the connection between the 3D model and the real world. Most users recognized buildings and features in an intuitive and natural manner, but for her the process seemed slow and awkward.

Why did this subject behave in this manner?

The most obvious explanation is that our system design contains one or more flaws that makes it difficult to use. This is a very real possibility, and one that could conceivably not be caught up by our experiment due to the uniform subject demographics, low number of participants or other factors that we failed to take into consideration. This subject's performance was one of the primary finds that spoke against the viability of M3D as we have implemented it.

A very interesting hypothesis that could have a potential impact on the design and use of M3D was suggested to us: It bases itself on individual differences in working with different communication methods. Some people are apparently unable to use common 2D maps. Like dyslectics have problems establishing the connection between letters and the sounds they represent, these people find the lines and colours of maps illegible. Doubtlessly, these are extremes on a scale as there are degrees of natural aptitude among individuals in almost every field. Perhaps this applies to individuals using real time 3D maps as well. Perhaps our subject could be considered to have little “talent” for 3D navigation, or the other subjects were all naturally adept at it. Though an interesting hypothesis, verifying its accuracy would take a lengthy study in itself - one we must leave to other researchers.

11.7 Further work

Four closely related research for further work is identified:

- **Investigating 3D graphics and the effect of constructing “mental maps”.**
  During our research we did not give enough consideration to the construction of mental maps by the subjects. Instead of referencing the 3D model while walking, the subjects memorized parts of it and used their memory as reference during task completion. The ability of users to create a mental map of 3D data may be of great value for future M3D systems. Incorporating mental mapping into design in a greater degree may facilitate this process and improve usability.

- **Improve data collection during experiment to further investigate dual- vs. multi-tasking.**
  Our experiment could have benefited from a more structured form of data collection. This experiment revealed the basic use-pattern of the subjects. This information could be used in another experiment design to allow for more fine-grained data collection to create a more detailed model of dual- vs multi-tasking in M3D.

- **Investigating alternative input/output techniques of terminals. Finding improved solutions for mobile 3D interaction.**
  The stylus- and touchscreen-interface appeared less than ideal to navigate 3D worlds. Several new PDA and hybrid PDA/mobile phones offer alternative interaction techniques. Investigating these techniques and suggesting new forms of interacting with M3D worlds could be interesting.

- **Investigating alternative solutions for annotations in M3D systems**
  In our prototype system we did not implement a feature that was a central part of our design idea: annotations. While we included a form of added information through the GUI-objects, Infocube/screen, Golden Path and Compass, these were very limited. We experimented with several different design ideas for implementing annotations for features inside the 3D world, but did not include any of these in the final design. We decided that we would focus on the general feasibility of M3D before attacking specific implementation issues like this. We postulated that the method used on the Infocube/screen, layering of tasks, would be applicable for other annotations as well. By

clicking a separate icon, the user could signal that he wanted to select a feature from his current viewpoint. When clicking on the object, the annotation information for this object would be displayed. This approach solves many of the problems presented by annotations inside an M3D system, and testing it against other approaches could be interesting.

Chapter 12

Conclusion

OUR belief that *Real time 3D on mobile terminals will be demonstrably useful: it will provide new capabilities or make the user more efficient* is strengthened after finalizing this research project. The nature of 3D presents information to a user in ways that other mediums can't capture; such as the shape of objects and the possibililty to display any viewpoint imaginable. This was evident in our experiment where the subjects took advantage of these 3D features to help orientate themselves and locate their destinations. That the users quickly learned how to operate the system and that they used the inherent advantages of the RT3D model (unlimited viewpoints, realism, high level of detail and motion) to accomplish their tasks indicates that M3D can be utilized effectively.

At the beginning of this thesis we formulated two separate research questions:

1. **How can a mobile system be designed to compensate for shared attention?**
   Identifying these methods relies on practical experiments, as the subjects use of mental maps illustrates. That the subjects reacted positively to the interaction features that aided navigation and orientation point to that such features have a positive effect. Three of our finds can be directly applied to M3D system design:

   (a) *Quick and easy navigation is of paramount importance in truly mobile use.* The stylus- and touchscreen-interface was deemed insufficient for M3D navigation.

   (b) *There is a need for a more intuitive and simple means of navigating 3D worlds while mobile.* Partial automation of navigation appeared to be a possible solution.

   (c) *Different navigation methods appeal to different users.* A full system should include several navigation modes, of which ours are examples, to allow different users to select the one they're most comfortable with.

2. **Can users utilize our system?**
   In general, we observed that the subjects did actively utilize our system as a tool in performing the tasks they were given. However, our results leave two key issues unanswered: To what *degree* the subjects utilized the system and whether they actually shared their attention between the system and the real world.
   The first problem stemmed from our explorative research approach. We were mainly interested in if and how an M3D system could be useful, whereas to determine to what degree the system is useful/usable one would have to perform a comparative study with other tools.
   The second problem was caused by our original way of thinking and conceptualizing shared attention. Our results can be read to indicate that aiming for "dual-tasking" as the goal of a shared attention system is not appropriate. In other words, one should perhaps not deal with shared attention by creating a system that is designed to be used while the user remains mobile. The use-pattern observed in our test subjects suggests that incorporating both "high attention" and "low attention" functions could be a better solution. Our results indicate that altering the design by grouping the features into these categories would allow the user to take advantage of the system while incorporating a dynamically changing use pattern.

Chapter 13

Reflections on the development process

STARTING from scratch with little experience in developing mobile or 3D systems, we often found ourselves faced with unfamiliar challenges. The VRML standard set constraints on our design choices, forcing us to use the Pocket Cortona VRML browser (which was the only VRML browser available for the Pocket PC at the time) This system came with its own predefined sets of navigation and GUI-possibilities. Modelling the campus area was almost trivial with the proper tools such as 3D Studio Max and Cosmoworlds. The time consuming element in modelling the wireframe model was tweaking it to have a low polygon count and still present enough detail in the full system (with textures) to support quick recognition. Furthermore, taking photographs of the buildings and modifying these for use as textures required more time and effort than we assumed.

Once we were satisfied with the realism, the level of detail and rendering speed of the 3D system, we found that using this model in a truly mobile situation was far from trivial - we needed a GUI and automatic navigation aids. GUI-design and VRML script-coding took a unexpected amount of time but this was required to develop a functional prototype for testing.

During this development phase we used a highly experimental approach that gave us valuable experience. Especially annotations were considered. In a full system annotations of some sort are likely to be a central feature. Implementing several different types of annotations (from yellow cubes hovering outside buildings and objects, to small clickable icons attached to top of buildings and objects) gave unsatisfactory results. We found that the annotations drew too much attention and they also obscured one of our system requirements - making the model realistic.

Another dilemma was that when navigating the 3D world using the stylus, one could easily activate the annotation unintentionally. In the final version of the prototype system we omitted these features, as testing them would have made for a separate thesis. We surmised that a menu-system and layering the task of selecting features for annotation information (in the same way that the info-screen was “layered” inside the info-cube) would be a viable solution in a complete system.

Another design possibility was tested. We considered dividing the model into smaller sub-areas and linking them to each other using VRML hyperlinks. Having smaller portions of Blindern Campus stored in the PDA memory certainly aided performance. Unfortunately the Pocket Cortona browser needed several minutes loading each cell’s 3D model and textures, making transitions between the sub-areas tedious. We tried to remedy this by designing the experiment so that the system could load the next task while the subject walked from the target of the last task and back to the starting point.

The iterative design process was carried out at one of the university labs that we shared with other students. In presenting the system for our peers at different stages we got valuable feedback on new design ideas and received help fine-tuning aspects of the design. Several visits to the Octaga company for feedback and guidance also helped us in making design choices and finding technical solutions.

Bibliography

Allison, R. S., Harris, L. R., Jenkin, M., Pintilie, G., Redlick, F. & Zikovitz, D. C. (2000), ‘First steps with a rideable computer’.
*http://www.cs.yorku.ca/ jenkin/papers/2000/vr00.pdf

Almstrum, V. L. (2003), Internet. Visited january 2003.
*http://www.cs.utexas.edu/users/almstrum/cs370/elvisino/rules.html

Ames, A. L. (2003), Internet. Visited january 2003.
*http://www.webtechniques.com/archives/1997/07/vrml/

Azuma, R. (1995), ‘A survey of augmented reality’.
*http://citeseer.nj.nec.com/article/azuma97survey.html

Azuma, R. (1999), ‘The challenge of making augmented reality work outdoors’.
*http://citeseer.nj.nec.com/azuma99challenge.html

Bahl, P. & Padmanabhan, V. N. (2000), RADAR: An in-building RF-based user location and tracking system, *in* ‘INFOCOM(2)’, pp. 775–784.
*http://citeseer.nj.nec.com/bahl00radar.html

Baum, D. (1998*a*), Internet. Visited july 2002.
*http://www.siggraph.org/publications/newsletter/v32n1/contributions/baum.html

Baum, D. (1998*b*), ‘3d graphics hardware: where we have been, where we are now, and where we are going’, *ACM SIGGRAPH Computer Graphics* **32**(1), 65–66.

Beck, E. T., Christiansen, M. K. & Kolbe, N. (2002), ‘Metoder til brugbarhedstest af mobile apparater’, *Kandidat oppgave, Aalborg University* .
*http://www.cs.auc.dk/library/cgi-bin/detail.cgi?id=1023094360

Beyer, H. & Holtzblatt, K. (1998), *Contextual design: Defining Customer Centered Systems*, Morgan Kaufmann, San Francisco, USA.

Beyer, H. & Holtzblatt, K. (1999), ‘Contextual design’, *interactions* **6**(1), 32–42.

Bluetooth.org (2002), Internet. Visited july, 2002.
*http://www.bluetooth.org/

Borälva, E., Göranssonb, B., Olssona, E. & Sandblada, B. (n.d.), ‘Usability and efficiency. the helios approach to development of user interfaces.’.

Brachtl, M., Slajs, J. & Slavík, P. (2000), ‘Pda based navigation system for a 3d environment’.
*http://citeseer.nj.nec.com/458058.html

Brown, B. & Perry, M. (2001), ‘Of maps and guidebooks: designing geographical technologies’, *ACM SIGGROUP Bulletin* **22**(3), 28–32.

Butz, A., Baus, J., Krüger, A. & Lohse, M. (2001), A hybrid indoor navigation system, *in* ‘IUI2001: International Conference on Intelligent User Interfaces’, ACM, New York.
*http://www.iuiconf.org/01pdf/2001-001-0005.pdf

Carroll, J., Mack, R. & Kellogg, W. (1988), ‘Interface metaphors and user interface design: Handbook of human-computer interaction’, *M. Helander, Elsevier Science Publisher B.V. (North Holland)* pp. 67–85.

Clark, J. J. (1999), ‘Attention and x-reality’.
*http://www.cim.mcgill.ca/ clark/papers/jjclark_iris_1999.pdf

Cyber Town (2002), Internet. Visited october 2002.
*http://www.cybertown.com

DirectX (2002), Internet. Visited june 2002.
*http://www.microsoft.com/windows/directx/default.asp

DJuice (2003), Internet. Visited january 2003.
*http://djuice.com

Duvaas, L. (2002), ‘Visualisering av grafalgoritmer’.
*http://www.stud.ifi.uio.no/ larsd/graf/

Edelman, S. & Weinshall, D. (1994), *Computational approaches to shape constancy*, Cambridge University Press.
*http://citeseer.nj.nec.com/edelman96computational.html

Eysenck, M. W. & Keane, M. T. (1995), *Cognitive Psychology*, Lawrence Erlbaum Associates Ltd., London.

Eysenck, M. W. (1984), *A Handbook of Cognitive Psychology*, Lawrence Erlbaum Associates Ltd., Publishers, London.

Falk, J. & Björk, S. (2001), ‘Crossbreeding wearable and ubiquitous computing: Design experiences from the bubblebadge’.
*http://citeseer.nj.nec.com/falk01crossbreeding.html

Faulkner, C. (1998), *The essence of Human-Computer Interaction*, Ashford Colour Press Ltd.

Feiner, S., MacIntyre, B. & Höllerer, T. (1997), ‘A touring machine: Prototyping 3d mobile augmented reality for exploring the urban environment’.

Finn karttjeneste (2002), Internet. Visited october 2002.
*http://www.finn.no/finn/map

Foley, J. D., Dam, A. V. & Feiner, S. K. (1994), *Introduction to Computer Graphics*, Addison-Wesley Publishing Company.

Fotonor AS (2002), Internet. Visited december 2002.
*http://www.fotonor.no/

Games Application Programming Interface (2003), Internet. Visited january 2003.
*http://www.microsoft.com/mobile/developer/technicalarticles/gapi.asp

Harter, A., Hopper, A., Steggles, P., Ward, A. & Webster, P. (1999), The anatomy of a context-aware application, *in* ‘Mobile Computing and Networking’, pp. 59–68.
*http://citeseer.nj.nec.com/harter02anatomy.html

Hollerer, T., Pavlik, J. V. & Feiner, S. (1999), Situated documentaries: Embedding multimedia presentations in the real world, *in* ‘ISWC’, pp. 79–86.
*http://citeseer.nj.nec.com/hollerer99situated.html

Holloway, R. L. (1995), Registration errors in augmented reality systems, Technical Report TR95-016.
*http://citeseer.nj.nec.com/holloway95registration.html

Hom, J. (1996), Internet.
*http://jthom.best.vwh.net/usability/thnkalod.htm

Institute of Telecommunications Sciences (1996), Internet.
*http://www.its.bldrdoc.gov/fs-1037/dir-023/_3346.htm

IT-avisen (2002), Internet. Visited june 2002.
*http://www.itavisen.no/art/1297008.html

Jameson, A., Schafer, R., Weis, T., Berthold, A. & Weyrath, T. (1999), Making systems sensitive to the user’s time and working memory constraints, *in* ‘Intelligent User Interfaces’, pp. 79–86.
*http://citeseer.nj.nec.com/jameson99making.html

Kirste, T., Herfet, T. & Schnaider, M. (2001), ‘Embassi: Multimodal assistance for universal access to infotainment and service infrastructures’, *ACM* pp. 42, 43.

Kitt Peak Inc. (2001), Internet. Visited august 2001.
*http://www.kittpeak.co.jp/racingdays/index_e.html

Koenen, R. (1993), *Usability Engineering*, AP Professional, Boston.

Kray, C. & Porzel, R. (2000), 'Spatial cognition and natural language interfaces in mobile personal assistants'.
*http://citeseer.nj.nec.com/kray00spatial.html

Kristoffersen, S. & Ljungberg, F. (1998*a*), 'Representing modalities in mobile computing'. IMC'98, Interactive applications of Mobile Computing, International workshop.
*http://www.rostock.igd.fhg.de/veranstaltungen/workshops/imc98/

Kristoffersen, S. & Ljungberg, F. (1998*b*), 'Your mobile computer is a stationary computer'.
*http://www.viktoria.se/publications/98/VRR-98-34.pdf

Köykkä, M., Ollikainen, R., Ranta-aho, M., Milszus, W., Wasserroth, S. & Friedrich, M. (1999), 'Usability heuristic guidelines for 3d multiuser worlds'.
*http://www.comlab.hut.fi/hft/publications/usability.pdf

Lauber, J. K., Bray, R. S., Harrison, R. L., Hemmingway, J. C. & Scott, B. C. (1982), 'An operational evaluation of head-up displays for civil transport operations (nasa technical paper 1815). moffett field, ca: Nasa ames research center'.

Liu, Z. & Kersten, D. (1998), 2D observers for human 3D object recognition?, *in* M. I. Jordan, M. J. Kearns & S. A. Solla, eds, 'Advances in Neural Information Processing Systems', Vol. 10, The MIT Press.
*http://citeseer.nj.nec.com/118230.html

Lowe, D. G. (1987), 'Three-dimensional object recognition from single two- dimensional images', *Artificial Intelligence* **31**(3), 355–395.
*http://citeseer.nj.nec.com/lowe87threedimensional.html

Lynch, K. (1960), *The image of the city*, The Technology Press & Harvard University Press.

Lähteenmäki, J., Laitinen, H. & Nordström, T. (n.d.).

Manovich, L. (2001), *The Language of New Media*, MIT press.

Marcus, A. (2001), Babyface design for mobile devices and the web, *in* 'Smith, Michael J., and Salvendy, Gavriel, Eds., Proceedings, Vol. 2, Human-Computer Interface Internat. (HCII)', pp. pp. 514–518.
*http://www.amanda.com/resources/HCII01/HCII01_Pn_Bab_Marcus_160401.pdf

Marine Doom (2002), Internet. Visited september 2002.
*http://www.tec.army.mil/TD/tvd/survey/Marine_Doom.html

McCann, R., Foyle, D. & Johnston, J. (1993), Attentional limitations with head-up displays, *in* 'Proceedings of the Seventh International Symposium on Aviation Psychology, Columbus: Ohio State University.', pp. 70–75.
*http://human-factors.arc.nasa.gov/ihi/hcsl/pubs/McCann_AvPsyc93.pdf

Moving Pictures Experts Group (Visited 2003), Internet.
*http://mpeg.telecomitalialab.com/standards/mpeg-4/mpeg-4.htm

MPEG Commitee website (2001), Internet. Visited january 2003.
*http://mpeg.telecomitalialab.com/

Myers, B. A. (93), Why are Human-Computer Interfaces Difficult to Design and Implement?, Technical Report CMU-CS-93-183.
*http://citeseer.nj.nec.com/myers93why.html

Myers, M. D. & Avison, D. (2002), *Qualitative Research in Information Systems*, 1 edn, SAGE publications.

Neumann, U. & Majoros, A. (2002), Internet. Visited Sept, 2002.
*http://www.dsls.usra.edu/dsls/meetings/shfp/proceedings/NeumannMajorospresent.pdf

Nielsen, J. (1994), 'Guerrilla hci: Using discount usability engineering to penetrate the intimidation barrier'.
*http://www.useit.com/papers/guerrilla_hci.html

Norman, D. A. (1988), *The Design of Everyday Things*, MIT Press.

Octaga AS (2003), Internet. Visited during 2000-2003.
*http://www.octaga.com

Open GL - www.opengl.org (2002), Internet. Visited june 2002.
*http://www.opengl.org

Overview of the VRML2.0 manual (1996), Internet. Visited Sept 5th, 2002.
*http://www.web3d.org/resources/vrml_ref_manual/

Pascoe, J., Ryan, N. S. & Morse, D. (2000), 'Using while moving: Hci issues in fieldwork environments', *ACM Transactions on Computer Human Interaction* **7**, 417–437.
*http://www.cs.ukc.ac.uk/pubs/2000/1346

Pascoe, J. & Ryan, N. (1999), Internet.
*http://www.cs.ukc.ac.uk/projects/mobicomp/Fieldwork/index.html

PCWorld.no (2002), Internet. Visited august 2002.
*http://www.pcworld.no/dataguiden/maskinvare/grafikkort/3d.shtm

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., & Carey, T. (1994), *Human-Computer Interaction*, Addison-Wesley.

Preece, J., Rogers, Y. & Sharp, H. (2002), *Interaction Design: Beyond human-computer interaction*, John Wiley & sons Inc.

Rakkolainen, I., Timmerheid, J. & Vainio, T. (2000), ‘A 3d city info for mobile users’.
*http://citeseer.nj.nec.com/rakkolainen00city.html

Reitmayr, G. & Schmalstieg, D. (2001), ‘Mobile collaborative augmented reality’.
*http://citeseer.nj.nec.com/reitmayr01mobile.html

Rhodes, B. J., Minar, N. & Weaver, J. (1999), ‘Wearable computing meets ubiquitous computing: Reaping the best of both worlds’, *ISWC* pp. 141–149.
*http://citeseer.nj.nec.com/rhodes99wearable.html

Schnädelbach, H., Koleva, B., Flintham, M., Fraser, M. & Chandler, P. (2002), ‘The augurscope: A mixed reality interface for outdoors’.
*http://citeseer.nj.nec.com/schndelbach01augurscope.html

Shneiderman, B. (1996), The eyes have it: A task by data type taxonomy for information visualizations, *in* ‘Proceedings of the IEEE Conference on Visual Languages’, pp. pp336–343.

Shneiderman, B. (1997), *Designing the User Interface: Strategies for Effective Human-Computer Interaction, third ed.*, Addison-Wesley.

Silicon Graphics (2002), Open inventor, Technical report, SGI. Visited Sept 5th, 2002.
*http://www.sgi.com/software/inventor/

Silverman, D. (2001), *Interpreting Qualitative Data: Methods for Analyzing Talk, Text and Interaction 2nd ed.*, SAGE Publications Ltd.

Skog, O.-J. (1998), *Å forklare sosiale fenomener*, Ad notam Gyldendal.

SMIL - Synchronized Multimedia Integration Language (2001), Internet. Visited july, 2001.
*http://www.w3.org/AudioVideo

Solberg, M. (2001), ‘Developing wireless communication services: The future is in the air’.
*http://www.ifi.uio.no/ maikens/thesis.pdf

Sony-Ericsson (2003), Internet. Visited january 2003.
*http://www.sonyericsson.com/no/spg.jsp?page=start

Spelke, E. S., Hirst, W. & Neisser, U. (1976), ‘Skills of divided attention’.

Statistisk Sentralbyrå (2003), Internet. Visited january 2003.
*http://www.ssb.no/aarbok/tab/t-101250-523.html

Strayer, D. L. & Johnston, W. A. (2001), 'Driven to distraction: dual-task studies of simulated driving and conversing on a cellular telephone', *Psychological Science* **12**(6), 462–466.

TINA organization (2002), Internet. Visited Sept 8th, 2002.
*http://www.tina-c.org

Vinson, N. G. (1999), Design guidelines for landmarks to support navigation in virtual environments, *in* 'CHI', pp. 278–285.
*http://citeseer.nj.nec.com/vinson99design.html

VOXEL-MAN 3D-Navigator: Brain and Skull (2002), Internet. Visited november 2002.
*http://www.uke.uni-hamburg.de/institute/imdm/idv/vm3dn/brainandskull.en.html

VRML2.0 spesification (1996), Internet. Visited Sept 5th, 2002.
*http://www.vrml.org/VRML2.0/FINAL/spec/part1/introduction.html

Väänänen-Vainio-Mattila, K. & Ruuska, S. (1998), 'User needs for mobile communication devices: Requirements gathering and analysis through contextual inquiry, in c. johson, ed., 'proceedings of the first workshop on human computer interaction with mobile devices".
*http://www.dsc.gla.ac.uk/johnson/papers/mobile/HCIMD1.html

White, S. (2002), Internet. Visited september 2002.
*http://www.ox.compsoc.net/ swhite/history/timeline-GAMES.html

Wideberg, K. (2001), *Historien om et kvalitativt forskningsprosjekt*, Universitetsforlaget.

World Wide Web Consortium (2003), Internet. Visited january 2003.
*http://www.w3.org/

X3D - eXtensible3D (2001), Internet. Visited july, 2001.
*http://www.web3d.org/

You, S. (2002), Internet. Visited september 2002.
*http://deimos.usc.edu/ suyay/Author/Author.html

Zyda, M. & Sheehan, J. (1997), National Academy Press, Washington, D.C.

Appendix A

VRML code

A.1 3Dworld frame file

```
#VRML V2.0 utf8
###############################################
#
# Use this file as frame and incorporate task
# definition as specified
#
# VRML WORLD: Blindern Campus
# developed by Marko Steffensen &
# Terje Torma, 2002
#
# Permission to use and modify this code for academic
# purposes is granted
#
###############################################
WorldInfo {
  title "University of Oslo VRML modelling project"
  info ["(C) Copyright 2002 Marko Steffensen &
Terje Torma"]
 }
NavigationInfo {
  avatarSize [0.25,2.9, 1.3]
  speed 4.5
  headlight FALSE
 }
DEF Background01 Background {
```

```
  skyColor [0.24 0.21 0.61,
    0.51 0.49 0.91,
    0.89 0.91 0.95]
  skyAngle [0.79,1.57]
  groundColor [0.7 0.4 0.9]
 }
#----- insert viewpoints definitions -----#

#----- insert walkthroughs definitions -----#

#===== start general GUI =====#
#Lager fancy effekt for Informasjonsknappen.
DEF ROTATETIMER TimeSensor {
  cycleInterval 2
  loop TRUE
 }
#Locator registers the location and rotation of all
#objects ROUTED to it within 600m.
DEF Locator ProximitySensor {
  size 6000 6000 6000
  center 0 0 0
 }

#The screens are members of the group Stationaryobjects
#that are ROUTED to Locator.
DEF Stationaryobjects Transform {
  children [
  DEF ramme Transform {
   translation 0 0 20
   children [
   Shape {
    appearance Appearance{
     material Material {
      transparency 0.5
      }
     }

    geometry DEF rammen IndexedFaceSet {
     ccw TRUE
     solid FALSE
     coord DEF rammen-COORD Coordinate {
      point [-0.4 0.43 0,
     0.4 0.43 0,
     -0.4 -0.43 0,
```

```
     0.4 -0.43 0]
      }
     coordIndex [1, 0, 3, -1, 2, 3, 0, -1]
     }
    }
   ]
   }

  DEF bigscreen Transform {
   translation 0 0 20
   children [
   Shape {
    appearance Appearance {
     texture ImageTexture {
      url "redn.jpg"
      }
     }

    geometry Text {
    #----- insert infotext definition-----#
     fontStyle FontStyle {
      size 0.12
      family "SANS"
      style "Bold"
      justify "BEGIN"
      }
     }
    }
   ]
   }

  #Bare infobox roterer, fordi bare den har en
  #orientationinterpolator..
  DEF infobox Transform {
   translation 0.25 -0.3 -0.9
   children [
   DEF SPINNER OrientationInterpolator {
    key [0, 0.25,0.5, 0.75,1]
    keyValue [0 1 0 0,
      0 1 0 1.57,
      0 1 0 3.14,
      0 1 0 4.71,
      0 1 0 6.28
     ]
```

```
    }
   Shape {
    appearance Appearance {
     texture ImageTexture {
      url "info.jpg"
      }
     }
    geometry Box {size 0.06 0.06 0.06}
   }
  DEF boxtouch TouchSensor { }
  ]
  }
 #infobox2 is a copy of infobox and uses the same
 #timer etc. It is used as a stand-in for infobox
 #when the infobox appears in the upper left corner
 #of infoscreen. Why? Read the comments for the
 #scripts Shrink and Grow.
 DEF infobox2 Transform {
  translation 0.35 -0.3 20
  children [
  DEF SPINNER2 OrientationInterpolator {
   key [0, 0.25,0.5, 0.75,1]
   keyValue [0 1 0 0,
     0 1 0 1.57,
     0 1 0 3.14,
     0 1 0 4.71,
     0 1 0 6.28
    ]
   }
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "info.jpg"
     }
    }
   geometry Box { size 0.06 0.06 0.06
   }
  }
 DEF boxtouch2 TouchSensor { }
 ]
 }
 ]
}
```

```
#Locator registers the changes of position and orientation
#of the user and moves the Stationaryobjects to compensate.
ROUTE Locator.position_changed TO
Stationaryobjects.set_translation
ROUTE Locator.orientation_changed TO
Stationaryobjects.set_rotation
ROUTE ROTATETIMER.fraction_changed TO
SPINNER.set_fraction
ROUTE SPINNER.value_changed TO
infobox.set_rotation
ROUTE ROTATETIMER.fraction_changed TO
SPINNER2.set_fraction
ROUTE SPINNER2.value_changed TO
infobox2.set_rotation

#The Shrink script moves the infoscreen to the back (making it
#invisible to the user because of Locator) and (appearantly) the
#infobox to the lower right. In reality it moves
#infobox2 to the back, making it invisible, and moves infobox to
#the front, making it visible. Clicking on infobox while in this
#state activates the Grow script.
DEF Shrink Script {
  eventIn SFBool bigtouchIsActive
  eventOut SFVec3f moveinfobox
  eventOut SFVec3f moveinfobox2
  eventOut SFVec3f moveinfoscreen
  eventOut SFVec3f movebigscreen
  url "vrmlscript:
  function initialize () {
   }
  function bigtouchIsActive(active) {
   moveinfobox[0] = 0.25;
   moveinfobox[1] = -0.3;
   moveinfobox[2] = -0.9;

   moveinfobox2[0] = 0.25;
   moveinfobox2[1] = -0.3;
   moveinfobox2[2] = 20;

   moveinfoscreen[0] = 0;
   moveinfoscreen[1] = 0;
   moveinfoscreen[2] = 20;

   movebigscreen[0] = 0;
```

```
    movebigscreen[1] = 0;
    movebigscreen[2] = 20;
    }
   "
  }

#The Grow script moves the infoscreen and infobox2 to
#the front and infobox to the back. Clicking on infobox2
while in this state activates the Shrink script.
DEF Grow Script {
  eventIn SFBool smalltouchIsActive
  eventOut SFVec3f moveinfobox
  eventOut SFVec3f moveinfobox2
  eventOut SFVec3f moveinfoscreen
  eventOut SFVec3f movebigscreen
  url "vrmlscript:
  function initialize () {
   }
  function smalltouchIsActive(active) {
   moveinfobox[0] = 0.35;
   moveinfobox[1] = -0.3;
   moveinfobox[2] = 20;

   moveinfobox2[0] = -0.25;
   moveinfobox2[1] = 0.29;
   moveinfobox2[2] = -0.9;

   moveinfoscreen[0] = 0;
   moveinfoscreen[1] = 0;
   moveinfoscreen[2] = -1.1;

   movebigscreen[0] = -0.33;
   movebigscreen[1] = 0.28;
   movebigscreen[2] = -1;
   }
  "
 }

#These ROUTERs routes the touchsensors to the correct
#scripts and the scripts' eventOuts to the correct
#infosumtin.translations.
ROUTE boxtouch2.isActive TO Shrink.bigtouchIsActive
ROUTE boxtouch.isActive TO Grow.smalltouchIsActive
```

```
ROUTE Shrink.moveinfobox TO infobox.set_translation
ROUTE Shrink.moveinfobox2 TO infobox2.set_translation
ROUTE Shrink.moveinfoscreen TO ramme.set_translation
ROUTE Shrink.movebigscreen TO bigscreen.set_translation

ROUTE Grow.moveinfobox TO infobox.set_translation
ROUTE Grow.moveinfobox2 TO infobox2.set_translation
ROUTE Grow.moveinfoscreen TO ramme.set_translation
ROUTE Grow.movebigscreen TO bigscreen.set_translation

############################################################
# compass.wrl V1.0 (01-Oct-1998)
# =====================
# Idea spark : David Ott, TECFA, University of Geneva
# Conception : David Ott
# Copyright : David.Ott@tecfa.unige.ch
#
# Free for all non-commercial uses if you keep this notice
# intact and share with other.
#
# Description :
# The compass points to the center of the scene [0 0 0].
# Just inline it in your scene : Inline { url "compass.wrl" }
#
# modified by Marko Steffensen & Terje Torma, 2002
#############################################################

#############
# DashBoard #
#############
Group {
  children [
  DEF PS ProximitySensor {size 1e25 1e25 1e25}
  DEF T Transform {
   children [
   Collision {
    collide FALSE
    children [
    DEF compass Transform {
     translation -0.35 -0.35 -1.2 #lower middle
     scale .0015 .0015 .0015
     children [
     DEF pin Transform { # compass pin
     children [
```

```
     Transform {
      children [DEF Compass Transform {
       rotation 1 0 0 1.57
       children [

       Shape {
        appearance Appearance {
         material Material {
          diffuseColor 0.5 0 0
          emissiveColor 0.8 0 0
          }
         }
        geometry DEF Box01-FACES IndexedFaceSet {
         ccw TRUE
         solid TRUE
         coord DEF Box01-COORD Coordinate {
  point [
 -37.59 -2.34 26.59,32.12 -2.34 26.59,
 -3.02 -2.34 -26.59,-37.59 -11.84 26.59,
 32.12 -11.84 26.59,-3.02 -11.84 -26.59,
 -18.14 -11.84 26.59,-18.03 -2.34 26.59,
 12.48 -11.84 26.59,12.12 -2.34 26.59,
 -18.14 -11.84 99.57,-18.03 -2.34 99.57,
 12.48 -11.84 99.57,12.12 -2.34 99.57
]
         }
        texCoord DEF Box01-TEXCOORD TextureCoordinate {
  point [
 -0.00 0.89,1.00 0.89,0.61 0.94,
 -0.00 0.15,1.00 0.15,0.49 0.25,
 0.28 0.15,0.28 0.89,0.72 0.15,
 0.71 0.89,0.28 -3.95e-005, 0.28 0.75,
 0.72 0.00,0.72 0.75,0.05 0.59,
 1.01 0.87,0.54 1.18,0.32 0.78,
 0.74 0.79,0.27 0.73,0.76 0.73,
 0.43 0.70,0.43 0.93,0.66 0.70,
 0.66 0.93,0.32 0.94,0.56 0.94,
 0.32 0.70,0.56 0.70,0.41 0.94,
 0.61 0.75,0.41 0.75,0.27 0.16,
 0.76 0.1643]
        }
       coordIndex [
   7, 2, 0, -1, 6, 3, 5, -1,
   1, 9, 4, -1, 3, 6, 0, -1,
```

```
   2, 1, 5, -1, 4, 5, 1, -1,
   0, 2, 3, -1, 5, 3, 2, -1,
   6, 7, 0, -1, 8, 6, 5, -1,
   10, 12, 11, -1, 9, 2, 7, -1,
   12, 13, 11, -1, 4, 8, 5, -1,
   8, 4, 9, -1, 1, 2, 9, -1,
   7, 6, 10, -1, 10, 11, 7, -1,
   6, 8, 12, -1, 12, 10, 6, -1,
   9, 7, 11, -1, 11, 13, 9, -1,
   8, 9, 13, -1, 13, 12, 8, -1
  ]
       texCoordIndex [
      17, 16, 14, -1, 6, 3, 5, -1,
      1, 9, 4, -1, 3, 6, 0, -1,
      26, 25, 28, -1, 27, 28, 25, -1,
      2, 29, 30, -1, 31, 30, 29, -1,
      6, 7, 0, -1, 8, 6, 5, -1,
      21, 23, 22, -1, 18, 16, 17, -1,
      23, 24, 22, -1, 4, 8, 5, -1,
      8, 4, 9, -1, 15, 16, 18, -1,
      7, 6, 10, -1, 10, 11, 7, -1,
      6, 8, 12, -1, 12, 10, 6, -1,
      20, 19, 32, -1, 32, 33, 20, -1,
      8, 9, 13, -1, 13, 12, 8, -1
     ]
       }
      }
     ]
     }
    ]
    }
   ]
   }
  ]
  }
 ]
 }
 ]
 }
 ]
}

#This script animates the compass & the pin of the compass
DEF pinRot Script {
```

```
    eventIn SFVec3f changePinOrientation
    field SFNode orient USE PS
    eventIn SFRotation changeCompassOrientation
    eventOut SFRotation pinOrientation
    eventOut SFRotation compassOrientation
    url [
    "vrmlscript:
    function changeCompassOrientation(newOrientation) {
     newOrientation.angle = (newOrientation.angle*-1);
     compassOrientation = new SFRotation(newOrientation.x,
     newOrientation.y,
     newOrientation.z,
     newOrientation.angle);
     }
    function changePinOrientation(globalPosition) {
     globalPosition.x = -globalPosition.x;
     globalPosition.y = -globalPosition.y;
     globalPosition.z = -globalPosition.z;
     var actualVec = new SFVec3f(0,1,0);
     pinOrientation = new SFRotation(actualVec,globalPosition);
     }"
    ]
   }

  #dashboard routing
  ROUTE PS.position_changed TO T.set_translation
  ROUTE PS.orientation_changed TO T.set_rotation
  #compass routing for position_changed
  ROUTE PS.position_changed TO pinRot.changePinOrientation
  ROUTE pinRot.pinOrientation TO pin.set_rotation
  #compass routing for orientation_changed
  ROUTE PS.orientation_changed TO pinRot.changeCompassOrientation
  ROUTE pinRot.compassOrientation TO compass.set_rotation
  #===== end general GUI =====#

  #===== start world building =====#
  #----- insert world spesific objects definitions -----#
  DEF Frederikke Transform {
    translation -83.5 -9.33 1281
    scale 4.38 4.38 4.38
    children [
    Shape {
     appearance Appearance {
      texture ImageTexture {
```

```
     url "fredrikke.jpg"
     }
    }
   geometry DEF Frederikke-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Frederikke-COORD Coordinate {
      point [
     5.71 0.00 11.29,-5.71 0 -11.29,
     5.71 0 -11.29,-5.71 2.67 11.29,
     5.71 2.67 11.29,-5.71 2.67 -11.29,
     5.71 2.67 -11.29,-5.71 0.00 11.29
    ]
    }
   texCoord DEF Frederikke-TEXCOORD TextureCoordinate {
     point [
    0.19 0.02,0.30 -0.18,0.00 -0.18,
    0.00 0.99,0.19 0.99,0.30 0.62,
    0.00 0.62,0.00 0.02,0.52 0.22,
    0.52 0.22,0.52 0.22,0.52 0.22,
    0.19 0.02,1.00 0.01,0.19 1.01,1.00 1
   ]
   }
  coordIndex [
      3, 4, 6, -1, 6, 5, 3, -1,
      0, 2, 6, -1, 6, 4, 0, -1,
      2, 1, 5, -1, 5, 6, 2, -1,
      7, 0, 4, -1, 4, 3, 7, -1
     ]
  texCoordIndex [
 8, 9, 11, -1, 11, 10, 8, -1,
 12, 13, 15, -1, 15, 14, 12,
 -1, 2, 1, 5, -1, 5, 6, 2, -1,
 7, 0, 4, -1, 4, 3, 7, -1
]
  }
 }
 ]
}
DEF Idrettsbygningen Transform {
  translation -80.1 -4.78 1197
  scale 4.38 4.38 4.38
  children [
  Shape {
```

```
     appearance Appearance {
      texture ImageTexture {
       url "IDRETTSBYGNING.jpg"
       }
      }
     geometry DEF Idrettsbygningen-FACES IndexedFaceSet {
      ccw TRUE
      solid TRUE
      coord DEF Idrettsbygningen-COORD Coordinate {
       point [
      -7.00 0 5.06,7.00 0 5.06,
      -7.00 0 -5.06,7.00 0 -5.06,
      -7.00 2.12 5.06,7.00 2.12 5.06,
      -7.00 2.12 -5.06,7.00 2.12 -5.06
     ]
      }
      texCoord DEF Idrettsbygningen-TEXCOORD TextureCoordinate {
       point [
      0.25 0.01,0.81 0.01,
      1.00 0.52,0.00 0.53,
      0.84 0.22,0.98 0.22,
      0.99 0.99,0.01 0.99,
      0.00 0.02,0.81 0.01,
      0.01 0.53,0.81 0.52,
      0.25 0.51,0.81 0.51,
      0.84 0.36,0.98 0.3602
     ]
       }
      coordIndex [
 4, 5, 7, -1, 7, 6, 4, -1,
 0, 1, 5, -1, 5, 4, 0, -1,
 1, 3, 7, -1, 7, 5, 1, -1,
 3, 2, 6, -1, 6, 7, 3, -1
         ]
      texCoordIndex [
     4, 5, 15, -1, 15, 14, 4, -1,
     0, 1, 13, -1, 13, 12, 0, -1,
     8, 9, 11, -1, 11, 10, 8, -1,
     3, 2, 6, -1, 6, 7, 3, -1
    ]
      }
     }
    ]
  }
```

```
DEF Frederikke_mellombygg Transform {
  translation -81.23 -4.18 1226
  scale 4.38 4.38 4.38
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "mellombygg.jpg"
     }
    }
   geometry DEF Frederikke_mellombygg-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Frederikke_mellombygg-COORD Coordinate {
     point [
    2.33 0 1.30,2.33 0 -1.61,
    -2.33 1.58 1.30,2.33 1.58 1.30,
    -2.33 1.58 -1.61,2.33 1.58 -1.61
   ]
     }
    texCoord DEF Frederikke_mellombygg-TEXCOORD TextureCoordinate {
     point [
    0.01 0.01,0.99 0.01,0.27 0.85,
    0.77 0.85,0.27 0.96,0.77 0.96,
    0.01 0.80,0.99 0.80
   ]
     }
    coordIndex [
2, 3, 5, -1, 5, 4, 2, -1,
0, 1, 5, -1, 5, 3, 0, -1
       ]
    texCoordIndex [
   2, 3, 5, -1, 5, 4, 2, -1,
   0, 1, 7, -1, 7, 6, 0, -1
  ]
    }
   }
  ]
 }
DEF Vilhem_Bjerknes__hus Transform {
  translation 57.83 -6.75 1237
  scale 4.38 4.38 4.38
  children [
  Shape {
```

```
   appearance Appearance {
    texture ImageTexture {
     url "sophuslie.jpg"
     }
    }
   geometry DEF Vilhem_Bjerknes__hus-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Vilhem_Bjerknes__hus-COORD Coordinate {
     point [
    -9.99 0 4.67,9.99 0 4.67,
    -9.99 0 -4.67,9.99 0 -4.67,
    -9.99 3.40 4.67,9.99 3.40 4.67,
    -9.99 3.40 -4.67,9.99 3.40 -4.67
   ]
     }
    texCoord DEF Vilhem_Bjerknes__hus-TEXCOORD TextureCoordinate {
     point [
    0.00 0.36,1.00 0.36,
    0.58 0.00,0.87 0.74,
    0.00 0.74,1.00 0.73,
    0.00 0.00,0.00 0.74,
    0.75 0.12,0.90 0.12,
    0.75 0.27,0.90 0.27,
    0.87 0.99,0.00 0.99,
    0.58 0.36,0.00 0.3521
   ]
     }
    coordIndex [
4, 5, 7, -1, 7, 6, 4, -1,
0, 1, 5, -1, 5, 4, 0, -1,
3, 2, 6, -1, 6, 7, 3, -1,
2, 0, 4, -1, 4, 6, 2, -1
       ]
    texCoordIndex [
   8, 9, 11, -1, 11, 10, 8, -1,
   0, 1, 5, -1, 5, 4, 0, -1,
   7, 3, 12, -1, 12, 13, 7, -1,
   6, 2, 14, -1, 14, 15, 6, -1
  ]
    }
   }
  ]
 }
```

```
DEF Niels_Henrik_Abels_hus Transform {
  translation 98.13 -7.35 1295
  scale 4.38 4.38 4.38
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "mattebygg.jpg"
     }
    }
   geometry DEF Niels_Henrik_Abels_hus-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Niels_Henrik_Abels_hus-COORD Coordinate {
     point [
    -2.59 0 6.88,2.59 0 6.88,
    -2.59 0 -6.88,2.59 0 -6.88,
    -2.59 22.57 6.88,2.59 22.57 6.88,
    -2.59 22.57 -6.88,2.59 22.57 -6.88
   ]
     }
    texCoord DEF Niels_Henrik_Abels_hus-TEXCOORD TextureCoordinate {
     point [
    0.22 0.03,0.53 0.03,0.55 0.03,0.26 0.03,
    0.86 0.08,0.86 0.08,0.86 0.08,0.86 0.08,
    0.98 0.03,0.01 0.03,0.93 0.98,0.05 0.98,
    0.22 0.98,0.53 0.98,0.55 0.98,0.26 0.98
   ]
    }
    coordIndex [
4, 5, 7, -1, 7, 6, 4, -1,
0, 1, 5, -1, 5, 4, 0, -1,
3, 2, 6, -1, 6, 7, 3, -1,
2, 0, 4, -1, 4, 6, 2, -1
       ]
    texCoordIndex [
   4, 5, 7, -1, 7, 6, 4, -1,
   0, 1, 13, -1, 13, 12, 0, -1,
   3, 2, 14, -1, 14, 15, 3, -1,
   9, 8, 10, -1, 10, 11, 9, -1
  ]
    }
   }
  ]
```

```
 }
DEF Sophus_Lies_Auditorium Transform {
  translation 56.69 -6.90 1337
  scale 4.38 4.38 4.38
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "auditorium.jpg"
     }
    }
   geometry DEF Sophus_Lies_Auditorium-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Sophus_Lies_Auditorium-COORD Coordinate {
     point [
    -4.28 0 4.54,-4.28 0 -4.54,
    4.28 0 -4.54,-4.28 3.48 4.54,
    4.28 3.48 4.54,-4.28 3.48 -4.54,
    4.28 3.48 -4.54
   ]
     }
    texCoord DEF Sophus_Lies_Auditorium-TEXCOORD TextureCoordinate {
     point [
    1 0, 1 0, 0 0, 0 0, 0.50 0.19,1 1, 0 1,
    0.51 0.19,0.50 0.19,0.51 0.19,1 1, 0 1
   ]
     }
    coordIndex [
3, 4, 6, -1, 6, 5, 3, -1,
2, 1, 5, -1, 5, 6, 2, -1,
1, 0, 3, -1, 3, 5, 1, -1
       ]
    texCoordIndex [
   4, 7, 9, -1, 9, 8, 4, -1,
   2, 1, 5, -1, 5, 6, 2, -1,
   3, 0, 10, -1, 10, 11, 3, -1
  ]
    }
   }
  ]
 }
DEF Akademika Transform {
  translation -48.31 -9.32 1367
```

```
  scale 4.38 4.38 4.38
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "akademika.jpg"
     }
    }
   geometry DEF Akademika-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Akademika-COORD Coordinate {
     point [
    8.04 0 5.32,-4.74 -1.261e-005 -5.32,
    8.04 0 -5.32,-4.74 3.47 5.32,
    8.04 3.47 5.32,-4.74 3.47 -5.32,
    8.04 3.47 -5.32,-4.74 -2.519e-005 5.32
   ]
     }
    texCoord DEF Akademika-TEXCOORD TextureCoordinate {
      point [
     0.67 0.02,0.66 0.02,1.00 0.02,1.00 0.66,
     0.66 0.65,0.66 0.65,1.00 0.66,1.00 0.01,
     0.65 0.32,0.65 0.32,0.65 0.32,0.65 0.32,
     0.66 0.02,0 0.02,0.66 0.99,0.00 0.98
    ]
    }
   coordIndex [
       3, 4, 6, -1, 6, 5, 3, -1,
       0, 2, 6, -1, 6, 4, 0, -1,
       2, 1, 5, -1, 5, 6, 2, -1,
       1, 7, 3, -1, 3, 5, 1, -1
      ]
   texCoordIndex [
  8, 9, 11, -1, 11, 10, 8, -1,
  0, 2, 6, -1, 6, 4, 0, -1,
  13, 12, 14, -1, 14, 15, 13, -1,
  1, 7, 3, -1, 3, 5, 1, -1
 ]
   }
  }
 ]
}
DEF Administrasjonbygningen Transform {
```

```
  translation -107.3 -9.31 1380
  scale 4.38 4.38 4.38
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "adminbygg.jpg"
     }
    }
   geometry DEF Administrasjonbygningen-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Administrasjonbygningen-COORD Coordinate {
      point [
     4.15 0 4.15,-4.15 0 -4.15,
     4.15 0 -4.15,-4.15 12.19 4.15,
     4.15 12.19 4.15,-4.15 12.19 -4.15,
     4.15 12.19 -4.15
    ]
    }
   texCoord DEF Administrasjonbygningen-TEXCOORD TextureCoordinate {
    point [
   0 0, 1 0, 0 0, 0.16 0.02,
   1.00 0, 1 1, 0 1,0.17 0.02,
   0.16 0.03,0.17 0.03,0 1, 1.00 1
  ]
    }
   coordIndex [
       3, 4, 6, -1, 6, 5, 3, -1,
       0, 2, 6, -1, 6, 4, 0, -1,
       2, 1, 5, -1, 5, 6, 2, -1
      ]
   texCoordIndex [
  3, 7, 9, -1, 9, 8, 3, -1,
  0, 4, 11, -1, 11, 10, 0, -1,
  2, 1, 5, -1, 5, 6, 2, -1
 ]
   }
  }
 ]
}
DEF Sophus_Bugges_hus Transform {
  translation -89.12 -4.78 1097
  scale 4.38 4.38 4.38
```

```
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "hf.jpg"
     }
    }
   geometry DEF Sophus_Bugges_hus-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Sophus_Bugges_hus-COORD Coordinate {
      point [
     -7.61 0 6.47,8.08 0 6.47,-7.61 0 -5.16,
     8.08 0 -5.16,-7.61 3.56 6.47,8.08 3.56 6.47,
     -7.61 3.56 -5.16,8.08 3.56 -5.155
    ]
    }
   texCoord DEF Sophus_Bugges_hus-TEXCOORD TextureCoordinate {
    point [
   0.00 0.00,0.00 0.34,0.81 0.68,
   -0.00 0.68,
   0.83 0.43,0.95 0.43,0.81 1, -0.00 1,
   1.00 0.00,0.00 0.34,1.00 0.34,
   0.75 0.35,0.00 0.67,0.75 0.68,0.83 0.55,
   0.95 0.5507
  ]
    }
   coordIndex [
       4, 5, 7, -1, 7, 6, 4, -1,
       0, 1, 5, -1, 5, 4, 0, -1,
       1, 3, 7, -1, 7, 5, 1, -1,
       3, 2, 6, -1, 6, 7, 3, -1
      ]
   texCoordIndex [
  4, 5, 15, -1, 15, 14, 4, -1,
  0, 8, 10, -1, 10, 9, 0, -1,
  1, 11, 13, -1, 13, 12, 1, -1,
  3, 2, 6, -1, 6, 7, 3, -1
 ]
   }
  }
 ]
}
DEF P_A__Munchs_hus Transform {
```

```
  translation -124.6 -4.71 996.7
  scale 4.38 4.38 4.38
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "sv4.jpg"
     }
    }
   geometry DEF P_A__Munchs_hus-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF P_A__Munchs_hus-COORD Coordinate {
      point [
     5.69 0 2.95,15.69 0 2.95,
     5.69 4.94 2.95,15.69 4.94 2.95,
     5.69 4.94 -2.46,15.69 4.94 -2.455
    ]
    }
   texCoord DEF P_A__Munchs_hus-TEXCOORD TextureCoordinate {
     point [
    0.33 0.02,0.98 0.02,0.55 0.95,0.57 0.95,
    0.55 0.97,0.57 0.97,0.34 0.93,0.96 0.9067
   ]
   }
  coordIndex [
      2, 3, 5, -1, 5, 4, 2, -1,
      0, 1, 3, -1, 3, 2, 0, -1
     ]
  texCoordIndex [
 2, 3, 5, -1, 5, 4, 2, -1,
 0, 1, 7, -1, 7, 6, 0, -1
]
  }
 }
 ]
}
DEF Niels_Treschovs_hus Transform {
  translation -108.5 -3.49 1035
  scale 4.38 4.38 4.38
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
```

```
     url "sv5.jpg"
     }
    }
   geometry DEF Niels_Treschovs_hus-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Niels_Treschovs_hus-COORD Coordinate {
      point [
     -2.56 0 6.38,2.21 0 6.38,2.21 0 -5.85,
     -2.56 17.51 6.38,2.21 17.51 6.38,
     -2.56 17.51 -5.85,2.21 17.51 -5.85
    ]
    }
   texCoord DEF Niels_Treschovs_hus-TEXCOORD TextureCoordinate {
     point [
    0.89 0.19,0.95 0.19,0.00 0.01,0.96 0.85,
    0.99 0.85,0.96 0.94,0.99 0.94,1.00 0.01,
    0.08 1.00,0.92 0.98,0.83 0.96,0.93 0.9632
   ]
   }
  coordIndex [
      3, 4, 6, -1, 6, 5, 3, -1,
      0, 1, 4, -1, 4, 3, 0, -1,
      1, 2, 6, -1, 6, 4, 1, -1
     ]
  texCoordIndex [
 3, 4, 6, -1, 6, 5, 3, -1,
 0, 1, 11, -1, 11, 10, 0, -1,
 2, 7, 9, -1, 9, 8, 2, -1
]
  }
 }
 ]
}
DEF Eilert_Sunds_hus_A_blokk Transform {
  translation 44.09 -4.75 1020
  scale 4.38 4.38 4.38
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "SV3.jpg"
     }
    }
```

```
   geometry DEF Eilert_Sunds_hus_A_blokk-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Eilert_Sunds_hus_A_blokk-COORD Coordinate {
      point [
     -16.28 0 3.44,1.28 0 3.44,
     -16.28 0 -3.78,-16.28 3.66 3.44,
     1.28 3.66 3.44,-16.28 3.66 -3.78,
     1.28 3.66 -3.783
    ]
    }
   texCoord DEF Eilert_Sunds_hus_A_blokk-TEXCOORD TextureCoordinate {
    point [
   1.00 0.53,0.00 0.03,0.57 0.53,0.99 1.00,
   0.11 0.71,0.57 1.00,0.25 0.71,1.00 0.02,
   0.01 0.49,1.00 0.48,0.11 0.86,0.25 0.8551
  ]
    }
   coordIndex [
       3, 4, 6, -1, 6, 5, 3, -1,
       0, 1, 4, -1, 4, 3, 0, -1,
       2, 0, 3, -1, 3, 5, 2, -1
      ]
   texCoordIndex [
  4, 6, 11, -1, 11, 10, 4, -1,
  1, 7, 9, -1, 9, 8, 1, -1,
  2, 0, 3, -1, 3, 5, 2, -1
 ]
   }
  }
 ]
}
DEF Eilert_Sunds_hus_B_blokk Transform {
  translation 62.35 -3.48 1064
  scale 4.38 4.38 4.38
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "svmain.jpg"
     }
    }
   geometry DEF Eilert_Sunds_hus_B_blokk-FACES IndexedFaceSet {
    ccw TRUE
```

```
    solid TRUE
    coord DEF Eilert_Sunds_hus_B_blokk-COORD Coordinate {
      point [
     -3.07 0 8.02,3.08 0 8.02,
     -3.07 0 -7.12,3.08 0 -7.12,
     -3.07 17.3 8.02,3.08 17.3 8.02,
     -3.07 17.3 -7.12,3.08 17.3 -7.119
    ]
    }
   texCoord DEF Eilert_Sunds_hus_B_blokk-TEXCOORD TextureCoordinate {
    point [
   0.03 0.23,0.15 0.22,0.46 0.04,0.22 0.04,
   0.83 0.09,0.84 0.09,0.83 0.10,0.84 0.10,
   0.98 0.04,0.02 0.04,0.91 0.98,0.08 0.99,
   0.08 0.99,0.18 0.99,0.46 1.00,0.22 0.9926
  ]
    }
   coordIndex [
       4, 5, 7, -1, 7, 6, 4, -1,
       0, 1, 5, -1, 5, 4, 0, -1,
       3, 2, 6, -1, 6, 7, 3, -1,
       2, 0, 4, -1, 4, 6, 2, -1
      ]
   texCoordIndex [
  4, 5, 7, -1, 7, 6, 4, -1,
  0, 1, 13, -1, 13, 12, 0, -1,
  3, 2, 14, -1, 14, 15, 3, -1,
  9, 8, 10, -1, 10, 11, 9, -1
 ]
   }
  }
 ]
}
DEF Eilert_Sunds_hus_C_blokk Transform {
  translation 27.97 -4.75 1120
  scale 4.38 4.38 4.38
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "cblokk.jpg"
     }
    }
   geometry DEF Eilert_Sunds_hus_C_blokk-FACES IndexedFaceSet {
```

```
    ccw TRUE
    solid TRUE
    coord DEF Eilert_Sunds_hus_C_blokk-COORD Coordinate {
      point [
     -3.44 0 8.59,3.44 0 8.59,-3.44 0 -8.59,
     3.44 0 -8.59,-3.44 4.81 8.59,3.44 4.81 8.59,
     -3.44 4.81 -8.59,3.44 4.81 -8.592
    ]
    }
   texCoord DEF Eilert_Sunds_hus_C_blokk-TEXCOORD TextureCoordinate {
    point [
   0.00 0.41,0.49 0.41,1.00 0.42,0.50 0.42,
   0.01 1.00,0.48 1.00,0.98 0.99,0.52 0.99,
   0.76 0.07,0.00 0.07,0.75 0.40,0.01 0.40,
   0.79 0.17,0.92 0.17,0.79 0.30,0.92 0.3035
  ]
    }
   coordIndex [
       4, 5, 7, -1, 7, 6, 4, -1,
       0, 1, 5, -1, 5, 4, 0, -1,
       3, 2, 6, -1, 6, 7, 3, -1,
       2, 0, 4, -1, 4, 6, 2, -1
      ]
   texCoordIndex [
  12, 13, 15, -1, 15, 14, 12, -1,
  0, 1, 5, -1, 5, 4, 0, -1,
  3, 2, 6, -1, 6, 7, 3, -1,
  9, 8, 10, -1, 10, 11, 9, -1
 ]
   }
  }
 ]
}
DEF Bakke Transform {
  translation -23.03 -12.28 1305
  scale 0.96 0.96 0.96
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "brostein.jpg"
     }
    textureTransform TextureTransform {
     center 0.5 0.5
```

```
  scale 130 150
  }
 }
geometry DEF Bakke-FACES IndexedFaceSet {
 ccw TRUE
 solid TRUE
 coord DEF Bakke-COORD Coordinate {
   point [
  -107.2 3.12 47.57,45.18 3.12 47.57,
  -105.8 3.12 -34.73,45.06 3.12 -34.74,
  52.26 5.74 -34.75,-92.2 5.74 -38.96,
  31.16 3.12 -34.73,31.16 5.75 -38.96,
  31.17 5.74 -34.73,52.33 5.71 19.75,
  45.15 3.12 19.78,138.5 9.34 -313.6,
  45.32 5.91 19.75,52.19 3.11 47.57,
  52.2 5.79 20.75,138.5 5.90 17.38,
  138.5 3.12 47.57,-45.3 7.84 -313.9,
  -107.2 3.11 98.76,138.1 3.11 98.76,
  38.88 5.75 -50.7,-92.2 5.74 -50.7,
  38.85 7.84 -58.3, -103.4 7.84 -58.3,
  138.5 5.90 -50.44,138.5 5.90 -62.3,
  138.5 5.89 -231,62.98 7.33 -231.7,
  -45.3 7.84 -252, -103.4 7.78 -227.4,
  46.08 9.28 -281.2, -103.4 9.27 -308.2,
  37.56 7.81 -232,37.56 7.81 -280.7,
  37.56 7.81 -313.9, 46.08 9.30 -250.9,
  37.56 7.81 -250.8, 46.09 9.31 -232.4,
  138.5 9.34 -231.6,-51.86 9.36 -308.7,
  -51.87 9.37 -252.1, -48.91 7.84 -240,
  -55.12 7.72 -239.9, -19.29 7.84 -138.7,
  -80.09 7.82 -140.6, -19.8 7.84 -230.2,
  -51.64 7.84 -230.9, -57.19 7.74 -231.1,
  135.6 6.83 -147.8, 45.57 7.58 -147.3,
  46.79 7.54 -100.5, 139.1 6.11 -100]
 }
texCoord DEF Bakke-TEXCOORD TextureCoordinate { point [
0.06 0.38,0.68 0.38,0.07 0.58,0.68 0.58,
0.71 0.58,0.13 0.59,0.63 0.58,0.63 0.59,
0.63 0.58,0.71 0.45,0.68 0.45,1.06 1.26,
0.68 0.45,0.71 0.38,0.71 0.45,1.06 0.46,
1.06 0.38,0.32 1.26,0.06 0.26,1.06 0.26,
0.66 0.62,0.13 0.62,0.66 0.64,0.08 0.64,
1.06 0.62,1.06 0.65,1.06 1.06,0.76 1.06,
0.32 1.11,0.08 1.05,0.69 1.18,0.08 1.25,
```

```
     0.65 1.06,0.65 1.18,0.65 1.26,0.69 1.11,
     0.65 1.11,0.69 1.06,1.06 1.06,0.29 1.25,
     0.29 1.11,0.30 1.08,0.28 1.08,0.42 0.84,
     0.17 0.84,0.42 1.06,0.29 1.06,0.27 1.06,
     1.05 0.86,0.69 0.86,0.69 0.74,1.06 0.7423]
     }
    coordIndex [
        2, 0, 6, -1, 6, 0, 3, -1,
        10, 3, 0, -1, 8, 4, 20, -1,
        12, 9, 10, -1, 1, 10, 0, -1,
        14, 9, 12, -1, 12, 1, 13, -1,
        13, 14, 12, -1, 24, 25, 20, -1,
        4, 9, 15, -1, 9, 14, 15, -1,
        13, 16, 15, -1, 15, 14, 13, -1,
        18, 19, 1, -1, 1, 0, 18, -1,
        13, 1, 19, -1, 16, 13, 19, -1,
        5, 20, 21, -1, 32, 43, 22, -1,
        23, 22, 44, -1, 20, 7, 8, -1,
        5, 7, 20, -1, 15, 20, 4, -1,
        15, 24, 20, -1, 25, 22, 20, -1,
        38, 11, 37, -1, 11, 35, 37, -1,
        11, 34, 30, -1, 49, 32, 50, -1,
        34, 17, 33, -1, 36, 17, 32, -1,
        11, 30, 35, -1, 33, 17, 36, -1,
        32, 17, 28, -1, 34, 33, 30, -1,
        29, 39, 31, -1, 31, 39, 17, -1,
        29, 40, 39, -1, 23, 44, 29, -1,
        41, 28, 40, -1, 44, 22, 43, -1,
        42, 41, 40, -1, 42, 40, 29, -1,
        32, 45, 43, -1, 41, 45, 28, -1,
        32, 28, 45, -1, 46, 45, 41, -1,
        46, 41, 42, -1, 47, 46, 42, -1,
        44, 47, 42, -1, 26, 27, 48, -1,
        48, 27, 49, -1, 27, 32, 49, -1,
        50, 32, 22, -1, 25, 50, 22, -1,
        25, 51, 50, -1]
    texCoordIndex [
   2, 0, 6, -1, 6, 0, 3, -1,
   10, 3, 0, -1, 8, 4, 20, -1,
   12, 9, 10, -1, 1, 10, 0, -1,
   14, 9, 12, -1, 12, 1, 13, -1,
   13, 14, 12, -1, 24, 25, 20, -1,
   4, 9, 15, -1, 9, 14, 15, -1,
   13, 16, 15, -1, 15, 14, 13, -1,
```

```
 18, 19, 1, -1, 1, 0, 18, -1,
 13, 1, 19, -1, 16, 13, 19, -1,
 5, 20, 21, -1, 32, 43, 22, -1,
 23, 22, 44, -1, 20, 7, 8, -1,
 5, 7, 20, -1, 15, 20, 4, -1,
 15, 24, 20, -1, 25, 22, 20, -1,
 38, 11, 37, -1, 11, 35, 37, -1,
 11, 34, 30, -1, 49, 32, 50, -1,
 34, 17, 33, -1, 36, 17, 32, -1,
 11, 30, 35, -1, 33, 17, 36, -1,
 32, 17, 28, -1, 34, 33, 30, -1,
 29, 39, 31, -1, 31, 39, 17, -1,
 29, 40, 39, -1, 23, 44, 29, -1,
 41, 28, 40, -1, 44, 22, 43, -1,
 42, 41, 40, -1, 42, 40, 29, -1,
 32, 45, 43, -1, 41, 45, 28, -1,
 32, 28, 45, -1, 46, 45, 41, -1,
 46, 41, 42, -1, 47, 46, 42, -1,
 44, 47, 42, -1, 26, 27, 48, -1,
 48, 27, 49, -1, 27, 32, 49, -1,
 50, 32, 22, -1, 25, 50, 22, -1,
 25, 51, 50, -1]
  }
 }
 ]
}
DEF Airbase Transform {
  translation -34.2 -7.88 1281
  rotation -1 0 0 -1.57
  scale 0.62 0.62 0.62
  children [
  Shape {
   appearance Appearance {
    material Material {
     diffuseColor 4 4 4
     emissiveColor 0.4 0.4 0.4
     }
    }
   geometry DEF Airbase-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Airbase-COORD Coordinate {
      point [
     -2.6 0 2.6, 2.6 0 2.6,
```

```
     -2.30 0.35 -4.10,2.30 0.35 -4.10,
     -2.6 6.14 2.6, 2.6 6.14 2.6,
     -2.30 5.79 -4.10,
     2.30 5.79 -4.102
    ]
    }
   coordIndex [
       0, 2, 3, -1, 3, 1, 0, -1,
       4, 5, 7, -1, 7, 6, 4, -1,
       1, 3, 7, -1, 7, 5, 1, -1,
       3, 2, 6, -1, 6, 7, 3, -1,
       2, 0, 4, -1, 4, 6, 2, -1
      ]
   }
  }
 ]
}
DEF Air Transform {
  translation -35.69 -0.66 1285
  rotation 0.02 -0.71 0.71 -3.19
  scale 0.62 0.62 0.62
  scaleOrientation 0.02 -0.06 1.00 -0.63
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "air.gif"
     }
    }
   geometry DEF Air-FACES IndexedFaceSet {
    ccw TRUE
    solid FALSE
    coord DEF Air-COORD Coordinate {
      point [
     -7.12 3.50 7.49,7.12 3.50 7.49,
     -7.12 3.50 -7.49,7.12 3.50 -7.49
    ]
    }
   texCoord DEF Air-TEXCOORD TextureCoordinate {
     point [
    1.00 -0.00,-0.00 -0.00,
    1.00 1.00,-0.00 1.00
   ]
   }
```

```
  coordIndex [0, 2, 3, -1, 3, 1, 0, -1]
  texCoordIndex [0, 2, 3, -1, 3, 1, 0, -1]
  }
 }
 ]
}
DEF Box10 Transform {
  translation 57.51 -7.01 1293
  scale 4.38 4.38 4.38
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "hermes.jpg"
     }
    }
   geometry DEF Box10-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Box10-COORD Coordinate {
      point [
     -2.26 0 3.05,-2.26 0 0.02,
     -2.26 0.94 3.05,-2.26 0.94 0.02
    ]
    }
   texCoord DEF Box10-TEXCOORD TextureCoordinate {
     point [0.87 0.27,0.12 0.27,0.87 0.61,0.13 0.64]
   }
  coordIndex [1, 0, 2, -1, 2, 3, 1, -1]
  texCoordIndex [1, 0, 2, -1, 2, 3, 1, -1]
  }
 }
 ]
}
DEF fredrikketrappbunn Transform {
  translation -23.03 -12.28 1310
  scale 0.96 0.96 0.96
  children [
  Shape {
   appearance Appearance {
    material Material {
     diffuseColor 0.5 0.5 0.5
     emissiveColor 0.6 0.6 0.6
     }
```

```
 }
geometry DEF fredrikketrappbunn-FACES IndexedFaceSet {
 ccw TRUE
 solid TRUE
 coord DEF fredrikketrappbunn-COORD Coordinate {
   point [
  45.18 3.1 47.57,-17.13 3.06 -39.15,
  45.05 3.06 -39.16,52.26 5.74 -39.17,
  -17.13 5.74 -39.16,31.16 3.06 -39.16,
  31.16 5.72 -45.31,31.16 7.45 -39.16,
  52.33 5.71 14.11,45.15 3.06 14.11,
  45.15 5.71 14.11,52.35 3.10 47.57,
  45.34 8.17 47.56,45.15 8.18 14.11,
  52.5 8.16 47.56,52.36 5.72 17.57,
  52.52 8.19 17.57,52.33 8.17 14.11,
  31.16 7.45 -45.31,52.26 7.45 -39.17,
  52.26 7.46 -45.31,52.26 5.72 -45.31,
  -17.16 5.72 -45.31,-37.23 5.72 -45.29,
  -37.26 5.74 -39.15,-37.27 3.06 -39.15,
  31.16 3.78 -41.41,-17.14 3.79 -41.42,
  31.16 4.8 -41.41,-17.14 4.80 -41.41,
  31.16 4.80 -43.5, -17.15 4.80 -43.5,
  31.16 5.72 -43.5, -17.15 5.72 -43.5,
  31.16 3.78 -39.16,-17.14 3.78 -39.15,
  49.7 5.74 -39.17,49.68 5.71 14.11,
  49.7 4.83 -39.17,49.68 4.81 14.11,
  47.21 4.83 -39.16,47.25 4.81 14.11,
  47.21 3.93 -39.16,47.25 3.90 14.11,
  45.05 3.93 -39.16,45.15 3.90 14.11
 ]
 }
coordIndex [
    6, 22, 33, -1, 1, 5, 34, -1,
    7, 34, 5, -1, 5, 36, 3, -1,
    3, 7, 5, -1, 8, 3, 36, -1,
    2, 9, 45, -1, 10, 8, 37, -1,
    0, 10, 9, -1, 12, 13, 10, -1,
    10, 0, 12, -1, 15, 11, 14, -1,
    14, 16, 15, -1, 17, 8, 10, -1,
    10, 13, 17, -1, 7, 18, 6, -1,
    7, 3, 19, -1, 20, 21, 6, -1,
    6, 18, 20, -1, 19, 3, 21, -1,
    21, 20, 19, -1, 7, 19, 20, -1,
    20, 18, 7, -1, 35, 4, 1, -1,
```

```
       23, 24, 4, -1, 4, 22, 23, -1,
       24, 25, 1, -1, 1, 4, 24, -1,
       35, 26, 27, -1, 7, 28, 26, -1,
       27, 26, 28, -1, 29, 4, 27, -1,
       27, 28, 29, -1, 7, 30, 28, -1,
       29, 28, 30, -1, 31, 4, 29, -1,
       29, 30, 31, -1, 7, 32, 30, -1,
       31, 30, 32, -1, 33, 4, 31, -1,
       31, 32, 33, -1, 7, 6, 32, -1,
       33, 32, 6, -1, 22, 4, 33, -1,
       1, 34, 35, -1, 7, 26, 34, -1,
       35, 34, 26, -1, 27, 4, 35, -1,
       8, 36, 37, -1, 5, 38, 36, -1,
       37, 36, 38, -1, 10, 37, 39, -1,
       37, 38, 39, -1, 5, 40, 38, -1,
       39, 38, 40, -1, 10, 39, 41, -1,
       39, 40, 41, -1, 5, 42, 40, -1,
       41, 40, 42, -1, 10, 41, 43, -1,
       41, 42, 43, -1, 5, 44, 42, -1,
       43, 42, 44, -1, 10, 43, 45, -1,
       43, 44, 45, -1, 5, 2, 44, -1,
       45, 44, 2, -1, 10, 45, 9, -1
      ]
   }
  }
 ]
}
DEF Bakgrunn_s_r_rst Transform {
  translation 109.4 -6.78 1255
  scale 1.66 1.66 1.66
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "mattewalkway.jpg"
     }
    }
   geometry DEF Bakgrunn_s_r_rst-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Bakgrunn_s_r_rst-COORD Coordinate {
      point [
     -0.04 0 6.80,-5.17 0 1.36,
     -0.04 9.54 6.80,-5.17 9.54 1.36
```

```
    ]
    }
   texCoord DEF Bakgrunn_s_r_rst-TEXCOORD TextureCoordinate {
     point [1 0, 0 0, 1 1, 0 1]
   }
  coordIndex [1, 0, 2, -1, 2, 3, 1, -1]
  texCoordIndex [1, 0, 2, -1, 2, 3, 1, -1]
  }
 }
 ]
}
DEF Bakgrunn__st Transform {
  translation 118.1 -5.78 1201
  scale 1.66 1.66 1.66
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "bkgrd3.jpg"
     }
    }
   geometry DEF Bakgrunn__st-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Bakgrunn__st-COORD Coordinate {
      point [
     -10.06 0.96 10.21,-24.75 0.79 -61.54,
     -10.06 9.39 10.21,-24.75 9.22 -61.54]
    }
   texCoord DEF Bakgrunn__st-TEXCOORD TextureCoordinate {
     point [1.00 0.00,0.00 0.00,1.00 1.00,0.00 0.10]
   }
  coordIndex [1, 0, 2, -1, 2, 3, 1, -1]
  texCoordIndex [1, 0, 2, -1, 2, 3, 1, -1]
  }
 }
 ]
}
DEF Bkgrunn_s_r_s_r_st Transform {
  translation 80.85 -10.83 1334
  rotation 0 1 0 -1.48
  scale 1.66 1.66 1.66
  children [
  Shape {
```

```
   appearance Appearance {
    texture ImageTexture {
     url "bkgrd5.jpg"
     }
    }
   geometry DEF Bkgrunn_s_r_s_r_st-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Bkgrunn_s_r_s_r_st-COORD Coordinate {
      point [-10.24 0.86 2.33,-5.66 0.86 -3.67,
     -10.24 11.77 2.33,-5.66 11.77 -3.67]
    }
   texCoord DEF Bkgrunn_s_r_s_r_st-TEXCOORD TextureCoordinate {
    point [1 0, 0 0, 1 1, 0 1]
    }
   coordIndex [1, 0, 2, -1, 2, 3, 1, -1]
   texCoordIndex [1, 0, 2, -1, 2, 3, 1, -1]
   }
  }
 ]
}
DEF Bakgrunn_vest Transform {
  translation -120.7 -4.97 1141
  rotation 0 -1 0 -3.14
  scale 1.66 1.66 1.66
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "bkg2.jpg"
     }
    }
   geometry DEF Bakgrunn_vest-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Bakgrunn_vest-COORD Coordinate {
      point [
     0.82 0 9.79,-5.69 0 -20.67,
     0.82 8.27 9.79,-5.69 8.27 -20.67
    ]
    }
   texCoord DEF Bakgrunn_vest-TEXCOORD TextureCoordinate {
    point [1 0, 0 0, 1 1, 0 1]
    }
```

```
   coordIndex [1, 0, 2, -1, 2, 3, 1, -1]
   texCoordIndex [1, 0, 2, -1, 2, 3, 1, -1]
   }
  }
 ]
}
DEF Bakgrunn_S_r Transform {
  translation 2.15 -9.39 1367
  rotation 0 1 0 -1.53
  scale 1.66 1.66 1.66
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "bkgrd7.jpg"
     }
    }
   geometry DEF Bakgrunn_S_r-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Bakgrunn_S_r-COORD Coordinate {
      point [
     -6.51 0 9.02,-5.17 0 -21.93,
     -6.51 13.32 9.02,-5.17 13.32 -21.93
    ]
    }
   texCoord DEF Bakgrunn_S_r-TEXCOORD TextureCoordinate {
     point [1 0, 0 0, 1.00 0.92,0 1]
   }
  coordIndex [1, 0, 2, -1, 2, 3, 1, -1]
  texCoordIndex [1, 0, 2, -1, 2, 3, 1, -1]
  }
 }
 ]
}
DEF Bakgrunn_nord Transform {
  translation -39.49 -5.92 995.7
  rotation 0 -1 0 -1.58
  scale 1.66 1.66 1.66
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "bkgrd1.jpg"
```

```
     }
    }
   geometry DEF Bakgrunn_nord-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Bakgrunn_nord-COORD Coordinate {
      point [
     -5.67 0 7.67,-8.67 0 -10.08,
     -5.67 12.94 7.67,-8.78 13.46 -10.08
    ]
    }
   texCoord DEF Bakgrunn_nord-TEXCOORD TextureCoordinate {
     point [1.00 0.00,0.00 0.00,1.00 1, -0.00 1.00]
   }
  coordIndex [1, 0, 2, -1, 2, 3, 1, -1]
  texCoordIndex [1, 0, 2, -1, 2, 3, 1, -1]
  }
 }
 ]
}
DEF Plen Transform {
  translation -21.52 -72.15 1194
  scale 1.66 1.66 1.66
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "gress.gif"
     }
    textureTransform TextureTransform {
     center 0.5 0.5
     scale 120 100
     }
    }
   geometry DEF Plen-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Plen-COORD Coordinate {
      point [
     -8.91 40.67 15.09,-15.08 40.67 11.52,
     -12.02 40.67 -12.87,-13.35 40.67 15.09,
     -8.91 40.67 -10.14,-60.7 40.67 -12.95,
     -60.7 40.67 -66.67,-12.05 40.67 -66.67,
     -5.93 40.67 -18.62,10.22 40.67 -18.62,
```

```
       -5.93 40.67 -66.93,10.22 40.67 -66.93,
       13.38 40.67 -18.62,19.55 40.67 -18.62,
       13.38 40.67 -66.65,19.55 40.67 -66.65,
       -5.93 40.67 15.09,10.22 40.67 15.09,
       -5.93 40.67 -15.09,10.22 40.67 -15.09,
       13.38 40.67 15.09,19.55 40.67 15.09,
       13.38 40.67 -15.09,19.55 40.67 -15.09,
       -19.12 40.67 25.82,-10.48 40.67 25.82,
       -19.12 40.67 20.77,-10.48 40.67 20.77,
       24.11 40.67 11.39,79.13 40.67 11.39,
       24.11 40.67 -19.58,79.13 40.67 -19.58,
       -12.22 40.67 -10.75,-15.08 40.67 -10.81
      ]
      }
     texCoord DEF Plen-TEXCOORD TextureCoordinate {
       point [
      0.37 0.12,0.33 0.15,0.35 0.42,0.34 0.12,
      0.37 0.39,0 0.42,0 1.00,0.35 1.00,0.39 0.48,
      0.51 0.48,0.39 1, 0.51 1, 0.53 0.48,0.57 0.48,
      0.53 1.00,0.57 1.00,0.39 0.12,0.51 0.12,
      0.39 0.44,0.51 0.44,0.53 0.12,0.57 0.12,
      0.53 0.44,0.57 0.44,0.30 0, 0.36 0, 0.30 0.05,
      0.36 0.05,0.61 0.16,1 0.16,0.61 0.49,
      1 0.49,0.35 0.39,0.33 0.40]
     }
    coordIndex [
        32, 33, 1, -1, 3, 0, 4, -1,
        5, 2, 7, -1, 7, 6, 5, -1,
        8, 9, 11, -1, 11, 10, 8, -1,
        12, 13, 15, -1, 15, 14, 12, -1,
        16, 17, 19, -1, 19, 18, 16, -1,
        20, 21, 23, -1, 23, 22, 20, -1,
        24, 25, 27, -1, 27, 26, 24, -1,
        28, 29, 31, -1, 31, 30, 28, -1
       ]
    texCoordIndex [
   32, 33, 1, -1, 3, 0, 4, -1,
   5, 2, 7, -1, 7, 6, 5, -1,
   8, 9, 11, -1, 11, 10, 8, -1,
   12, 13, 15, -1, 15, 14, 12, -1,
   16, 17, 19, -1, 19, 18, 16, -1,
   20, 21, 23, -1, 23, 22, 20, -1,
   24, 25, 27, -1, 27, 26, 24, -1,
   28, 29, 31, -1, 31, 30, 28, -1
```

```
]
  }
 }
 ]
}
DEF SV_betongskilt Transform {
  translation 40.16 -3.17 1045
  scale 4.38 4.38 4.38
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "SV-skilt.jpg"
     }
    }
   geometry DEF SV_betongskilt-FACES IndexedFaceSet {
    ccw TRUE
    solid FALSE
    coord DEF SV_betongskilt-COORD Coordinate {
      point [
     -2.26 0 3.05,-2.26 0 0.02,
     -2.26 0.94 3.05,-2.26 0.94 0.02
    ]
    }
   texCoord DEF SV_betongskilt-TEXCOORD TextureCoordinate {
    point [1 0, 0 0, 1 1, 0 1]
    }
   coordIndex [1, 0, 2, -1, 2, 3, 1, -1]
   texCoordIndex [1, 0, 2, -1, 2, 3, 1, -1]
   }
  }
 ]
}
DEF fredrikketrapptopp Transform {
  translation -90.41 -14.43 1263
  rotation -1 0 0 -1.59
  children [
  Shape {
   appearance Appearance {
    material Material {
     diffuseColor 0.5 0.5 0.5
     emissiveColor 0.6 0.6 0.6
     }
    }
```

```
   geometry DEF fredrikketrapptopp-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF fredrikketrapptopp-COORD Coordinate {
      point [
     32.19 -6.35 -9.54,104.4 -6.30 -7.51,
     32.16 -15.31 -9.38,104.4 -15.31 -9.38,
     48.58 -6.35 -9.54,48.57 -15.31 -9.38,
     48.58 -6.31 -7.51,32.04 -6.31 -7.51,
     104.4 -11.84 -8.77,48.57 -11.84 -8.77,
     104.4 -8.73 -8.82,48.56 -8.73 -8.82,
     104.4 -8.72 -8.19,48.56 -8.72 -8.19,
     104.4 -6.32 -8.24,48.58 -6.32 -8.24,
     104.4 -11.85 -9.44,48.57 -11.85 -9.44
    ]
    }
   coordIndex [
       0, 4, 2, -1, 14, 15, 1, -1,
       4, 5, 2, -1, 4, 6, 15, -1,
       6, 1, 15, -1, 0, 7, 6, -1,
       6, 4, 0, -1, 16, 17, 8, -1,
       9, 8, 17, -1, 4, 9, 17, -1,
       8, 9, 10, -1, 11, 10, 9, -1,
       4, 11, 9, -1, 10, 11, 12, -1,
       13, 12, 11, -1, 4, 13, 11, -1,
       12, 13, 14, -1, 15, 14, 13, -1,
       4, 15, 13, -1, 3, 5, 16, -1,
       17, 16, 5, -1, 4, 17, 5, -1
      ]
   }
  }
 ]
}
DEF Box27 Transform {
  translation -14.99 -3.76 1243
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "fountain.gif"
     }
    }
   geometry DEF Box27-FACES IndexedFaceSet {
    ccw TRUE
```

```
    solid FALSE
    coord DEF Box27-COORD Coordinate {
      point [
     -7.00 0 4.63,7.25 0 4.63,
     -7.00 2.89 4.63,7.25 2.89 4.63
    ]
    }
   texCoord DEF Box27-TEXCOORD TextureCoordinate {
     point [0 0, 1 0, 0 1, 1 1]
   }
  coordIndex [0, 1, 3, -1, 3, 2, 0, -1]
  texCoordIndex [0, 1, 3, -1, 3, 2, 0, -1]
  }
 }
 ]
}
DEF Auditorium_overbygg Transform {
  translation 44.49 -12.56 1287
  rotation -1 0 0 -3.14
  scale 4.38 4.38 4.38
  children [
  Shape {
   appearance Appearance {
    material Material {
     diffuseColor 0 0 0
     }
    }
   geometry DEF Auditorium_overbygg-FACES IndexedFaceSet {
    ccw FALSE
    solid FALSE
    coord DEF Auditorium_overbygg-COORD Coordinate {
      point [
     1.36 -1.31 -6.82,1.36 -1.31 6.89,
     1.37 -2.64 -6.82,1.37 -2.64 6.89,
     1.36 -2.50 6.12,1.36 -1.31 6.12,
     1.36 -2.48 4.29,1.36 -1.30 4.30,
     1.36 -2.47 4.19,1.36 -1.30 4.17,
     1.36 -2.48 1.62,1.36 -1.30 1.62,
     1.36 -2.48 1.50,1.36 -1.30 1.52,
     1.36 -2.46 -0.81,1.36 -1.31 -0.83,
     1.36 -2.48 -0.94,1.36 -1.31 -0.93,
     1.36 -2.46 -3.73,1.36 -1.30 -3.74,
     1.36 -2.47 -3.85,1.36 -2.46 -3.85,
     1.36 -1.30 -3.86,1.36 -2.46 -5.47,
```

```
    1.36 -1.31 -5.46,-1.36 -1.31 -6.82,
    -1.36 -1.31 6.89,-1.36 -2.64 -6.82,
    -1.36 -2.64 6.89,-1.36 -2.50 6.12,
    -1.36 -1.31 6.12,-1.36 -2.48 4.29,
    -1.36 -1.30 4.30,-1.36 -2.47 4.19,
    -1.36 -1.30 4.17,-1.36 -2.48 1.62,
    -1.36 -1.30 1.62,-1.36 -2.48 1.50,
    -1.36 -1.30 1.52,-1.36 -2.46 -0.81,
    -1.36 -1.31 -0.83,-1.36 -2.48 -0.94,
    -1.36 -1.31 -0.93,-1.36 -2.46 -3.73,
    -1.36 -1.30 -3.74,-1.36 -2.47 -3.85,
    -1.36 -2.46 -3.85,-1.36 -1.30 -3.86,
    -1.36 -2.46 -5.47,-1.36 -1.31 -5.47
   ]
   }
  coordIndex [
      24, 0, 2, -1, 2, 3, 4, -1,
      2, 23, 24, -1, 20, 23, 2, -1,
      4, 3, 1, -1, 5, 4, 1, -1,
      8, 6, 7, -1, 4, 6, 8, -1,
      7, 9, 8, -1, 4, 8, 10, -1,
      12, 10, 11, -1, 4, 10, 12, -1,
      11, 13, 12, -1, 4, 12, 14, -1,
      16, 14, 15, -1, 4, 14, 16, -1,
      15, 17, 16, -1, 4, 16, 18, -1,
      4, 18, 20, -1, 2, 4, 20, -1,
      20, 18, 21, -1, 21, 18, 19, -1,
      19, 22, 21, -1, 20, 21, 23, -1,
      27, 3, 2, -1, 3, 27, 28, -1,
      49, 27, 25, -1, 27, 29, 28, -1,
      27, 49, 48, -1, 45, 27, 48, -1,
      29, 26, 28, -1, 30, 26, 29, -1,
      33, 32, 31, -1, 29, 33, 31, -1,
      32, 33, 34, -1, 29, 35, 33, -1,
      37, 36, 35, -1, 29, 37, 35, -1,
      36, 37, 38, -1, 29, 39, 37, -1,
      41, 40, 39, -1, 29, 41, 39, -1,
      40, 41, 42, -1, 29, 43, 41, -1,
      29, 45, 43, -1, 27, 45, 29, -1,
      45, 46, 43, -1, 46, 44, 43, -1,
      44, 46, 47, -1, 45, 48, 46, -1]
  }
 }
]
```

```
}
DEF SVoverbygg Transform {
  translation 43.02 -9.10 1065
  rotation -1 0 0 -3.14
  scale 4.38 4.38 4.38
  children [
  Shape {
   appearance Appearance {
    material Material {
     diffuseColor 0 0 0
     }
    }
   geometry DEF SVoverbygg-FACES IndexedFaceSet {
    ccw FALSE
    solid FALSE
    coord DEF SVoverbygg-COORD Coordinate {
      point [
     1.36 -1.31 -6.82,1.36 -1.31 6.89,
     1.37 -2.64 -6.82,1.37 -2.64 6.89,
     1.36 -2.50 6.12,1.36 -1.31 6.12,
     1.36 -2.48 4.29,1.36 -1.30 4.30,
     1.36 -2.47 4.19,1.36 -1.30 4.17,
     1.36 -2.48 1.62,1.36 -1.30 1.62,
     1.36 -2.48 1.50,1.36 -1.30 1.52,
     1.36 -2.46 -0.81,1.36 -1.31 -0.83,
     1.36 -2.48 -0.94,1.36 -1.31 -0.93,
     1.36 -2.46 -3.73,1.36 -1.30 -3.74,
     1.36 -2.47 -3.85,1.36 -2.46 -3.85,
     1.36 -1.30 -3.86,1.36 -2.46 -5.47,
     1.36 -1.31 -5.46,-1.36 -1.31 -6.82,
     -1.36 -1.31 6.89,-1.36 -2.64 -6.82,
     -1.36 -2.64 6.89,-1.36 -2.50 6.12,
     -1.36 -1.31 6.12,-1.36 -2.48 4.29,
     -1.36 -1.30 4.30,-1.36 -2.47 4.19,
     -1.36 -1.30 4.17,-1.36 -2.48 1.62,
     -1.36 -1.30 1.62,-1.36 -2.48 1.50,
     -1.36 -1.30 1.52,-1.36 -2.46 -0.81,
     -1.36 -1.31 -0.83,-1.36 -2.48 -0.94,
     -1.36 -1.31 -0.93,-1.36 -2.46 -3.73,
     -1.36 -1.30 -3.74,-1.36 -2.47 -3.85,
     -1.36 -2.46 -3.85,-1.36 -1.30 -3.86,
     -1.36 -2.46 -5.47,-1.36 -1.31 -5.47
    ]
    }
```

```
    coordIndex [
        24, 0, 2, -1, 2, 3, 4, -1,
        2, 23, 24, -1, 20, 23, 2, -1,
        4, 3, 1, -1, 5, 4, 1, -1,
        8, 6, 7, -1, 4, 6, 8, -1,
        7, 9, 8, -1, 4, 8, 10, -1,
        12, 10, 11, -1, 4, 10, 12, -1,
        11, 13, 12, -1, 4, 12, 14, -1,
        16, 14, 15, -1, 4, 14, 16, -1,
        15, 17, 16, -1, 4, 16, 18, -1,
        4, 18, 20, -1, 2, 4, 20, -1,
        20, 18, 21, -1, 21, 18, 19, -1,
        19, 22, 21, -1, 20, 21, 23, -1,
        27, 3, 2, -1, 3, 27, 28, -1,
        49, 27, 25, -1, 27, 29, 28, -1,
        27, 49, 48, -1, 45, 27, 48, -1,
        29, 26, 28, -1, 30, 26, 29, -1,
        33, 32, 31, -1, 29, 33, 31, -1,
        32, 33, 34, -1, 29, 35, 33, -1,
        37, 36, 35, -1, 29, 37, 35, -1,
        36, 37, 38, -1, 29, 39, 37, -1,
        41, 40, 39, -1, 29, 41, 39, -1,
        40, 41, 42, -1, 29, 43, 41, -1,
        29, 45, 43, -1, 27, 45, 29, -1,
        45, 46, 43, -1, 46, 44, 43, -1,
        44, 46, 47, -1, 45, 48, 46, -1
       ]
    }
   }
  ]
 }
 DEF HFoverbygg Transform {
   translation -83.8 -10.25 1069
   rotation -0.71 0 -0.71 -3.14
   scale 4.38 4.38 4.38
   children [
   Shape {
    appearance Appearance {
     material Material {
      diffuseColor 4 4 4
      }
     }
    geometry DEF HFoverbygg-FACES IndexedFaceSet {
     ccw FALSE
```

```
    solid FALSE
    coord DEF HFoverbygg-COORD Coordinate {
      point [
     1.02 -1.31 -6.82,1.02 -1.31 6.89,
     1.03 -2.64 -6.82,1.03 -2.64 6.89,
     1.02 -2.50 6.76,1.02 -1.31 6.76,
     1.02 -2.48 4.29,1.02 -1.30 4.29,
     1.02 -2.47 4.19,1.02 -1.30 4.20,
     1.02 -2.48 1.62,1.02 -1.30 1.62,
     1.02 -2.48 1.50,1.02 -1.30 1.52,
     1.02 -2.46 -0.81,1.02 -1.31 -0.83,
     1.02 -2.48 -0.94,1.02 -1.31 -0.93,
     1.02 -2.46 -3.73,1.02 -1.30 -3.74,
     1.02 -2.47 -3.85,1.02 -2.46 -3.85,
     1.02 -1.30 -3.86,1.02 -2.46 -5.47,
     1.02 -1.31 -5.46,-1.32 -1.31 -6.82,
     -1.32 -1.31 6.89,-1.32 -2.64 -6.82,
     -1.32 -2.64 6.89,-1.32 -2.50 6.76,
     -1.32 -1.31 6.76,-1.32 -2.48 4.29,
     -1.32 -1.30 4.29,-1.32 -2.47 4.19,
     -1.32 -1.30 4.20,-1.32 -2.48 1.62,
     -1.32 -1.30 1.62,-1.32 -2.48 1.50,
     -1.32 -1.30 1.52,-1.32 -2.46 -0.81,
     -1.32 -1.31 -0.83,-1.32 -2.48 -0.94,
     -1.32 -1.31 -0.93,-1.32 -2.46 -3.73,
     -1.32 -1.30 -3.74,-1.32 -2.47 -3.85,
     -1.32 -2.46 -3.85,-1.32 -1.30 -3.86,
     -1.32 -2.46 -5.47,-1.32 -1.31 -5.47
    ]
    }
   coordIndex [
       24, 0, 2, -1, 2, 3, 4, -1,
       2, 23, 24, -1, 20, 23, 2, -1,
       4, 3, 1, -1, 5, 4, 1, -1,
       8, 6, 7, -1, 4, 6, 8, -1,
       7, 9, 8, -1, 4, 8, 10, -1,
       12, 10, 11, -1, 4, 10, 12, -1,
       11, 13, 12, -1, 4, 12, 14, -1,
       16, 14, 15, -1, 4, 14, 16, -1,
       15, 17, 16, -1, 4, 16, 18, -1,
       4, 18, 20, -1, 2, 4, 20, -1,
       20, 18, 21, -1, 21, 18, 19, -1,
       19, 22, 21, -1, 20, 21, 23, -1,
       27, 3, 2, -1, 3, 27, 28, -1,
```

```
        49, 27, 25, -1, 27, 29, 28, -1,
        27, 49, 48, -1, 45, 27, 48, -1,
        29, 26, 28, -1, 30, 26, 29, -1,
        33, 32, 31, -1, 29, 33, 31, -1,
        32, 33, 34, -1, 29, 35, 33, -1,
        37, 36, 35, -1, 29, 37, 35, -1,
        36, 37, 38, -1, 29, 39, 37, -1,
        41, 40, 39, -1, 29, 41, 39, -1,
        40, 41, 42, -1, 29, 43, 41, -1,
        29, 45, 43, -1, 27, 45, 29, -1,
        45, 46, 43, -1, 46, 44, 43, -1,
        44, 46, 47, -1, 45, 48, 46, -1
       ]
    }
   }
  ]
 }
 DEF Plass01 Transform {
   translation -23.03 -12.28 1304
   scale 0.96 0.96 0.96
   children [
   Shape {
    appearance Appearance {
     material Material {
      diffuseColor 0.4 0.4 0.4
      emissiveColor 0.5 0.5 0.5
      }
     }
    geometry DEF Plass01-FACES IndexedFaceSet {
     ccw TRUE
     solid TRUE
     coord DEF Plass01-COORD Coordinate {
       point [
      62.3 7.73 -232, 37.56 7.81 -232,
      37.56 7.81 -280.7, 61.95 10.48 -250.9,
      37.56 7.81 -250.8, 37.59 10.48 -250.8,
      37.57 10.45 -232, 62.13 10.46 -232,
      42.81 9.37 -280.7, 42.81 9.37 -250.8,
      40.62 9.40 -250.8, 40.63 9.40 -280.7,
      40.62 8.84 -250.8, 40.62 8.84 -280.7,
      39.2 8.84 -250.8, 39.2 8.84 -280.7,
      39.2 8.29 -250.8, 39.2 8.29 -280.7,
      37.56 8.29 -250.8, 37.56 8.29 -280.7,
      46.81 9.35 -250.9, 46.8 9.35 -280.7,
```

```
     62.13 7.69 -232, 61.95 9.25 -250.9
    ]
    }
   coordIndex [
       18, 2, 4, -1, 5, 1, 6, -1,
       7, 5, 6, -1, 0, 6, 1, -1,
       7, 6, 0, -1, 8, 9, 20, -1,
       3, 5, 7, -1, 20, 9, 5, -1,
       8, 11, 9, -1, 18, 4, 5, -1,
       5, 4, 1, -1, 11, 13, 10, -1,
       9, 10, 5, -1, 9, 11, 10, -1,
       13, 15, 12, -1, 10, 12, 5, -1,
       10, 13, 12, -1, 15, 17, 14, -1,
       12, 14, 5, -1, 12, 15, 14, -1,
       17, 19, 16, -1, 14, 16, 5, -1,
       14, 17, 16, -1, 19, 2, 18, -1,
       16, 18, 5, -1, 16, 19, 18, -1,
       8, 20, 21, -1, 3, 20, 5, -1,
       22, 23, 3, -1, 3, 7, 22, -1,
       23, 20, 3, -1
      ]
   }
  }
 ]
}
DEF Tr_r_Kantrad Transform {
  translation -17.52 7.92 1151
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "tree.gif"
     }
    }
   geometry DEF Tr_r_Kantrad-FACES IndexedFaceSet {
    ccw TRUE
    solid FALSE
    coord DEF Tr_r_Kantrad-COORD Coordinate {
      point [
     -9.63 -4.88 62.72,9.63 -4.88 62.72,
     -9.63 -4.88 -62.72,9.63 -4.88 -62.72,
     -9.63 -12.72 62.72,9.63 -12.72 62.72,
     -9.63 -12.72 -62.72,9.63 -12.72 -62.72
    ]
```

```
      }
     texCoord DEF Tr_r_Kantrad-TEXCOORD TextureCoordinate {
       point [
      1 1, 0.29 0.99,0 1, 0 1, 1 0, 0.56 0.93,0 0,
      1 1, 0.56 0.98,0.29 1.02,0.55 -0.06,0.29 -0.02,
      0 0, 1 0, 0.29 0.01,0.55 -0.04
     ]
     }
    coordIndex [
        1, 0, 5, -1, 4, 5, 0, -1,
        3, 1, 7, -1, 5, 7, 1, -1,
        2, 3, 6, -1, 7, 6, 3, -1,
        0, 2, 4, -1, 6, 4, 2, -1
       ]
    texCoordIndex [
 5, 1, 15, -1, 14, 15, 1, -1,
 7, 3, 13, -1, 12, 13, 3, -1,
 8, 9, 10, -1, 11, 10, 9, -1,
 0, 2, 4, -1, 6, 4, 2, -1
]
  }
 }
 ]
}
DEF Tr_r_Midtrad Transform {
  translation -27.1 7.92 1151
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "tree.gif"
     }
    }
   geometry DEF Tr_r_Midtrad-FACES IndexedFaceSet {
    ccw TRUE
    solid FALSE
    coord DEF Tr_r_Midtrad-COORD Coordinate {
      point [
     9.63 -4.88 62.72,9.63 -4.88 -62.72,
     9.63 -12.72 62.72,9.63 -12.72 -62.72
    ]
    }
   texCoord DEF Tr_r_Midtrad-TEXCOORD TextureCoordinate {
     point [0 1, 1 1, 0 0, 1 0]
```

```
    }
  coordIndex [1, 0, 3, -1, 2, 3, 0, -1]
  texCoordIndex [1, 0, 3, -1, 2, 3, 0, -1]
  }
 }
 ]
}
DEF Box28 Transform {
  translation -66.31 0 1039
  rotation 0 -1 0 -1.89
  children [
  Shape {
   appearance Appearance {
    texture ImageTexture {
     url "infoskilt.jpg"
     }
    }
   geometry DEF Box28-FACES IndexedFaceSet {
    ccw TRUE
    solid FALSE
    coord DEF Box28-COORD Coordinate {
      point [
     5.77 -3.64 -4.68,6.53 -3.64 -6.68,
     5.77 3.99 -4.68,6.53 3.99 -6.68
    ]
    }
   texCoord DEF Box28-TEXCOORD TextureCoordinate {
     point [0 0, 1 0, 0 1, 1 1]
   }
  coordIndex [0, 1, 3, -1, 3, 2, 0, -1]
  texCoordIndex [0, 1, 3, -1, 3, 2, 0, -1]
  }
 }
 ]
}
DEF Vann Transform {
  translation 24.79 -8.40 1072
  children [
  Shape {
   appearance Appearance {
    material Material {
     diffuseColor 0.2 0.2 0.2
     emissiveColor 0.2 0.1 0.5
     }
```

```
     }
    geometry DEF Vann-FACES IndexedFaceSet {
     ccw TRUE
     solid TRUE
     coord DEF Vann-COORD Coordinate {
       point [
      -10.92 6.17 10.19,10.92 6.17 10.19,
      -10.92 6.17 -6.35,10.92 6.17 -6.35]
     }
    coordIndex [0, 1, 3, -1, 3, 2, 0, -1]
    }
   }
  ]
 }
 DEF Bakgrunn_s_r_s_rvest Transform {
   translation -77.48 0 1409
   children [
   Shape {
    appearance Appearance {
     texture ImageTexture {
      url "bkgrd8.jpg"
      }
     }
    geometry DEF Bakgrunn_s_r_s_rvest-FACES IndexedFaceSet {
     ccw TRUE
     solid TRUE
     coord DEF Bakgrunn_s_r_s_rvest-COORD Coordinate {
       point [
      -11.69 14.44 -10.77,8.47 14.44 -18.7,
      -11.69 -9.38 -10.77,8.47 -9.38 -18.7]
     }
    texCoord DEF Bakgrunn_s_r_s_rvest-TEXCOORD TextureCoordinate {
     point [1 0.85,0 0.85,1 0, 0 0]
     }
    coordIndex [0, 1, 2, -1, 3, 2, 1, -1]
    texCoordIndex [0, 1, 2, -1, 3, 2, 1, -1]
    }
   }
  ]
 }
 DEF Bakgrunn_s_rvest Transform {
   translation -129.9 0 1345
   children [
   Shape {
```

```
   appearance Appearance {
    texture ImageTexture {
     url "bkgrd9.jpg"
     }
    }
   geometry DEF Bakgrunn_s_rvest-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Bakgrunn_s_rvest-COORD Coordinate {
      point [
     4.38 5.50 16.9, 25.4 5.50 -13.91,
     4.38 -9.38 16.9, 25.4 -9.40 -13.91
    ]
    }
   texCoord DEF Bakgrunn_s_rvest-TEXCOORD TextureCoordinate {
    point [0 0.89,1 0.89,0 0.00,1 0]
    }
   coordIndex [1, 0, 3, -1, 2, 3, 0, -1]
   texCoordIndex [1, 0, 3, -1, 2, 3, 0, -1]
   }
  }
 ]
}
DEF Box01 Transform {
  translation 70.21 -6.39 1300
  rotation 0.58 -0.58 0.58 -4.19
  children [
  Shape {
   appearance Appearance {
    material Material {
     diffuseColor 0.4 0.4 0.4
     emissiveColor 0.4 0.4 0.4
     }
    }
   geometry DEF Box01-FACES IndexedFaceSet {
    ccw TRUE
    solid TRUE
    coord DEF Box01-COORD Coordinate {
      point [
     -5.99 0 0.53,5.99 0 0.53,-5.99 0 -0.53,
     5.99 0 -0.53,-5.99 8.80 0.53,5.99 8.80 0.53,
     -5.99 8.80 -0.53,5.99 8.80 -0.53
    ]
    }
```

```
     coordIndex [
         0, 2, 3, -1, 3, 1, 0, -1,
         4, 5, 7, -1, 7, 6, 4, -1,
         0, 1, 5, -1, 5, 4, 0, -1,
         1, 3, 7, -1, 7, 5, 1, -1,
         3, 2, 6, -1, 6, 7, 3, -1,
         2, 0, 4, -1, 4, 6, 2, -1
        ]
     }
    }
   ]
  }
  DEF Busk Transform {
    translation 69.67 -5.55 1300
    rotation 0.58 -0.58 0.58 -4.19
    scale 0.89 0.89 0.89
    children [
    Shape {
     appearance Appearance {
      texture ImageTexture {
       url "busken.gif"
       }
      }
     geometry DEF Busk-FACES IndexedFaceSet {
      ccw TRUE
      solid TRUE
      coord DEF Busk-COORD Coordinate {
        point [
       -5.99 0 0.53,5.99 0 0.53,-5.99 0 -1.43,
       5.99 0 -1.43,-5.99 8.80 0.53,5.99 8.80 0.53,
       -5.99 8.80 -1.43,5.99 8.80 -1.43
      ]
      }
     texCoord DEF Busk-TEXCOORD TextureCoordinate {
       point [
      1 -0.01,0.00 -0.00,1.00 0.52,-0.00 0.53,
      1 0, 1 0, 1 0.53,1.00 0.53,0 0, 1 0, 0 0.54,
      1 0.54,0 0, 0 0.53,0 0, 0.00 0.53
     ]
     }
    coordIndex [
        0, 2, 3, -1, 3, 1, 0, -1,
        4, 5, 7, -1,7, 6, 4, -1,
        1, 3, 7, -1, 7, 5, 1, -1,
```

```
      2, 0, 4, -1, 4, 6, 2, -1
     ]
  texCoordIndex [
 0, 2, 3, -1, 3, 1, 0, -1,
 8, 9, 11, -1, 11, 10, 8, -1,
 5, 7, 15, -1, 15, 14, 5, -1,
 6, 4, 12, -1, 12, 13, 6, -1
]
  }
 }
 ]
}

DEF Box40 Transform {
  translation -69.33 -4.06 1062
  rotation 1 0 0 -1.57
  children [
  Shape {
   appearance Appearance {
    material Material {
     diffuseColor 0.4 0.4 0.4
     emissiveColor 0.4 0.4 0.4
     }
    }
   geometry DEF Box40-FACES IndexedFaceSet {
    ccw FALSE
    solid TRUE
    coord DEF Box40-COORD Coordinate {
      point [
     3.25 -10.84 -0.74,-3.64 -10.84 -0.81,
     3.25 -10.84 -0.23,-3.65 -10.84 0.73,
     3.25 26.82 -0.74,3.25 26.8 -0.23,
     -3.65 26.82 0.73,1.12 -10.84 0.24,
     1.12 26.8 0.24,-0.82 -10.84 0.74,
     -0.83 26.8 0.74,-0.82 -10.84 0.24,
     -0.82 26.8 0.24,1.12 -10.84 -0.23,
     1.12 26.8 -0.23,3.25 52.21 -0.74,
     3.25 52.21 -0.23,-3.65 52.21 0.73,
     3.25 34.7 -0.74,3.25 34.7 -0.23,
     -3.65 34.7 0.73,1.12 52.21 0.24,
     1.12 34.7 0.24,-0.82 52.21 0.74,
     -0.83 34.7 0.74,-0.82 52.21 0.24,
     -0.82 34.7 0.24,1.12 52.21 -0.23,
     1.12 34.7 -0.23,3.25 -12.64 -0.74,
```

```
      -3.64 -12.64 -0.81,-3.65 -12.64 0.73,
      -0.82 -12.64 0.74
     ]
     }
    coordIndex [
        2, 0, 13, -1, 5, 14, 4, -1,
        2, 13, 5, -1, 0, 2, 4, -1,
        5, 4, 2, -1, 7, 8, 14, -1,
        7, 0, 11, -1, 7, 11, 8, -1,
        8, 12, 4, -1, 9, 10, 12, -1,
        9, 3, 10, -1, 10, 6, 4, -1,
        3, 6, 10, -1, 11, 12, 8, -1,
        11, 0, 9, -1, 11, 9, 12, -1,
        12, 10, 4, -1, 13, 14, 5, -1,
        13, 0, 7, -1, 13, 7, 14, -1,
        14, 8, 4, -1, 16, 27, 15, -1,
        19, 18, 28, -1, 16, 19, 27, -1,
        15, 18, 16, -1, 19, 16, 18, -1,
        21, 28, 22, -1, 21, 25, 15, -1,
        21, 22, 25, -1, 22, 18, 26, -1,
        23, 26, 24, -1, 23, 24, 17, -1,
        24, 18, 20, -1, 17, 24, 20, -1,
        25, 22, 26, -1, 25, 23, 15, -1,
        25, 26, 23, -1, 26, 18, 24, -1,
        27, 19, 28, -1, 27, 21, 15, -1,
        27, 28, 21, -1, 28, 18, 22, -1,
        18, 6, 20, -1, 6, 18, 4, -1,
        1, 3, 31, -1, 31, 30, 1, -1,
        9, 0, 29, -1, 29, 32, 9, -1,
        3, 9, 32, -1, 32, 31, 3, -1
       ]
    }
   }
  ]
 }
 DEF Box02 Transform {
   translation -15.06 -7.70 1248
   children [
   Transform {
    translation 0 1.90 0
    children [
    Shape {
     appearance Appearance {
      material Material {
```

```
      diffuseColor 0.4 0.4 0.4
      emissiveColor 0.5 0.5 0.5
      }
     }
    geometry Box { size 13.72 3.79 11 }
    }
   ] }
   ]
  }
 DEF Box41 Transform {
   translation 70.21 -5.79 1300
   rotation 0.58 -0.58 0.58 -4.19
   children [
   Shape {
    appearance Appearance {
     texture ImageTexture {
      url "busken.gif"
      }
     }
    geometry DEF Box41-FACES IndexedFaceSet {
     ccw TRUE
     solid TRUE
     coord DEF Box41-COORD Coordinate {
       point [
      -4.45 1.33 -0.53,4.76 1.33 -0.53,
      -4.45 8.12 -0.53,4.76 8.12 -0.53
     ]
     }
    texCoord DEF Box41-TEXCOORD TextureCoordinate {
      point [0.89 0.14,0.12 0.17,0.83 0.43,0.18 0.42]
    }
   coordIndex [
   1, 0, 2, -1, 2, 3, 1, -1]
   texCoordIndex [
   1, 0, 2, -1, 2, 3, 1, -1]
   }
  }
 ]
}
#===== start world building =====#

#----- insert scripts definitions -----#
```

A.2 task0 definitions

```
# task0 definitions

#===== viewpoints definitions =====#
DEF Free Viewpoint {
   description "Free"
   position -43.61 -6.213 1320
}

DEF Groundpath Viewpoint {
   description "Groundpath"
   position -34.61 -6.213 1288
   orientation 0 1 0 -1.56
}

DEF Walkthrough Viewpoint {
  position -34.42 -6.26 1306
  orientation -0.832 -0.03628 0.5535 -0.03146
  description "Walkthrough"
}

DEF Birdseye Viewpoint {
  position -207.8 284.6 1339
  orientation 0.5469 0.7541 0.3637 -1.339
  description "Birdseye"
}
#===== end viewpoints definitions =====#

#===== walkthroughs definitions =====#
DEF Walkthrough-TIMER TimeSensor {
loop FALSE
cycleInterval 8 },

DEF Walkthrough-POS-INTERP PositionInterpolator {
  key [0.1, 0.2, 0.6, 0.7, 1]
  keyValue [-34.42 -6.26 1306,
    -34.42 -6.467 1288,
    16.1 -6.467 1288, 29.99
    -3.663 1288, 57.41 -3.665 1288]
}
DEF Walkthrough-ROT-INTERP OrientationInterpolator {
  key [0.1, 0.2 ]
  keyValue [0 -1 0 0, 0 -1 0 1.58,]
```

```
}

ROUTE Walkthrough-TIMER.fraction_changed TO
Walkthrough-POS-INTERP.set_fraction
ROUTE Walkthrough-POS-INTERP.value_changed TO
Walkthrough.set_position
ROUTE Walkthrough-TIMER.fraction_changed TO
Walkthrough-ROT-INTERP.set_fraction
ROUTE Walkthrough-ROT-INTERP.value_changed TO
Walkthrough.set_orientation
#===== end walkthroughs definitions =====#

#===== world spesific objects definitions =====#
DEF mattepath Transform {
  translation -33.09 12.81 1210
  rotation -0.001156 -1 -0.002909 -1.563
  scale 1.414 1.414 1.414
  scaleOrientation -0.008625 -0.02151 0.9997 -0.7856
  children [
    Shape {
      appearance Appearance {
        material Material {
  transparency 0.2 emissiveColor 0.8 0.8 0
}
      }
      geometry DEF mattepath-FACES IndexedFaceSet {
        ccw TRUE
        solid TRUE
        coord DEF mattepath-COORD Coordinate {
  point [
          -55.28 1.638 76.27, -59.45 1.66 76.09,
  -53.4 1.621 76.16, -49.83 1.604 76.35,
  -55.01 1.609 85.04, -63.18 -0.1217 -1.846,
          -63.14 -0.1131 0.3341, -56.56 -0.1312 0.2988,
  -54.48 -0.1478 -1.897, -56.22 1.713 41.43,
  -53.91 1.704 41.42, -54.34 -0.1834 21.65,
          -56.38 -0.2072 21.75, -54.22 -0.2091 36.73,
  -56.41 -0.1896 36.76, -53.76 1.684 62.33,
  -56.08 1.701 62.36]
        }
        coordIndex [
          2, 0, 3, -1, 1, 3, 0, -1, 3,
  1, 4, -1, 0, 2, 15, -1, 5, 7,
  8, -1, 8, 7, 12, -1, 10, 9, 16,
```

```
  -1, 6, 7, 5, -1, 12, 14, 11, -1,
  8, 12, 11, -1, 14, 9, 13, -1, 11,
  14, 13, -1, 13, 9, 10, -1, 0, 15,
  16, -1, 16, 15, 10, -1]
        }
    }
  ]
}

#===== end world spesific objects definitions =====#

#===== infotext definition =====#
string ["     Task 0"
        ""
        "Eksperimenter"
        "og bli kjent"
        "med systemet"]
#===== end infotext definition =====#

#===== scripts definitions =====#
DEF touchScript Script {
  eventIn SFBool view2IsActive
  eventOut SFVec3f settextcolor
  url "vrmlscript:
  function view2IsActive(active) {
    settextcolor[0] = -33.09;
    settextcolor[1] = -8.5;
    settextcolor[2] = 1210;
  }"
}

ROUTE Groundpath.isBound TO
touchScript.view2IsActive
ROUTE touchScript.settextcolor TO
mattepath.set_translation

DEF touchScript1 Script {
  eventIn SFBool view1IsActive
  eventOut SFVec3f settextcolor1
  url "vrmlscript:
  function view1IsActive(active) {
    settextcolor1[0] = -33.09;
    settextcolor1[1] = 13.09;
    settextcolor1[2] = 1210;
```

```
  }"
}

ROUTE Birdseye.isBound TO
touchScript1.view1IsActive
ROUTE Free.isBound TO
touchScript1.view1IsActive
ROUTE touchScript1.settextcolor1
TO mattepath.set_translation
DEF S Script {
  eventIn  SFTime bindTime
  eventOut SFTime startTime
  url "vrmlscript:
  function bindTime(t) {
    startTime = t;
  }"
}

ROUTE Walkthrough.bindTime TO S.bindTime
ROUTE S.startTime TO Walkthrough-TIMER.startTime
#====== end touchscript ########
```

A.3 task1 definitions

```
# task1 definitions

#===== viewpoints definitions =====#
DEF Free Viewpoint {
   description "Free"
   position -43.61 -6.213 1320
}

DEF Groundpath Viewpoint {
  description "Groundpath"
  position -32.61 -4.213 1300
  orientation 0 1 0 3.14
}

DEF Walkthrough Viewpoint {
  position -33.09 13.09 1345
  orientation 0.002887 1 -0.001147 -1.579
  description "Walkthrough"
}

DEF Birdseye Viewpoint {
  position 187.7 370.6 1500
  orientation 0.7928 -0.5745 -0.2035 -0.9861
  description "Birdseye"
}
#===== end viewpoints definitions =====#

#===== walkthroughs definitions =====#
DEF Walkthrough-TIMER TimeSensor {
loop FALSE
cycleInterval 12 }

DEF Walkthrough-POS-INTERP PositionInterpolator {
  key [0.1, 0.25, 0.5, 0.75 ]
  keyValue [-32.61 -4.213 1300,
    -32.83 -5.04 1336,
    -74.78 -5.04 1336,
    -70.23 -5.04 1380 ]
}

DEF Walkthrough-ROT-INTERP OrientationInterpolator {
  key [0.1, 0.2, 0.25, 0.45, 0.5, 0.75 ]
```

```
  keyValue [0 -1 0 -3.142, 0 -1 0 -3.142,
    0 -1 0 -1.58, 0 -1 0 -1.58,
    0 -1 0 -3.107, 0 -1 0 -1.58 ]
}


ROUTE Walkthrough-TIMER.fraction_changed TO
Walkthrough-POS-INTERP.set_fraction
ROUTE Walkthrough-POS-INTERP.value_changed TO
Walkthrough.set_position
ROUTE Walkthrough-TIMER.fraction_changed TO
Walkthrough-ROT-INTERP.set_fraction
ROUTE Walkthrough-ROT-INTERP.value_changed TO
Walkthrough.set_orientation
#===== end walkthroughs definitions =====#

#===== world spesific objects definitions =====#
DEF mattepath Transform {
  translation -33.09 12.81 1210
  rotation -0.001156 -1 -0.002909 -1.563
  scale 1.414 1.414 1.414
  scaleOrientation -0.008625 -0.02151 0.9997 -0.7856
  children [
    Shape {
      appearance Appearance {
        material Material {
  transparency 0.2 emissiveColor 0.8 0.8 0
}
      }
      geometry DEF mattepath-FACES IndexedFaceSet {
        ccw TRUE
        solid TRUE
        coord DEF mattepath-COORD Coordinate {
  point [
          -55.28 1.638 76.27, -59.45 1.66 76.09,
  -53.4 1.621 76.16, -49.83 1.604 76.35,
  -55.01 1.609 85.04, -63.18 -0.1217 -1.846,
          -63.14 -0.1131 0.3341, -56.56 -0.1312 0.2988,
  -54.48 -0.1478 -1.897, -56.22 1.713 41.43,
  -53.91 1.704 41.42, -54.34 -0.1834 21.65,
          -56.38 -0.2072 21.75, -54.22 -0.2091 36.73,
  -56.41 -0.1896 36.76, -53.76 1.684 62.33,
  -56.08 1.701 62.36]
        }
```

```
        coordIndex [
          2, 0, 3, -1, 1, 3, 0, -1, 3, 1, 4, -1,
  0, 2, 15, -1, 5, 7, 8, -1, 8, 7, 12, -1,
  10, 9, 16, -1, 6, 7, 5, -1, 12, 14, 11, -1,
  8, 12, 11, -1, 14, 9, 13, -1, 11, 14, 13, -1,
          13, 9, 10, -1, 0, 15, 16, -1, 16, 15, 10, -1]
        }
    }
  ]
}

#===== end world spesific objects definitions =====#

#===== infotext definition =====#
string ["     Task 1"
""
"Lokaliser og"
"ta deg frem til"
"administrasjons-"
"bygget."]
#===== end infotext definition =====#

#===== scripts definitions =====#
DEF touchScript Script {
  eventIn SFBool view2IsActive
  eventOut SFVec3f settextcolor
  url "vrmlscript:
  function view2IsActive(active) {
    settextcolor[0] = -33.09;
    settextcolor[1] = -9;
    settextcolor[2] = 1345;
  }"
}
ROUTE Groundpath.isBound TO touchScript.view2IsActive
ROUTE touchScript.settextcolor TO adminpath.set_translation
DEF touchScript1 Script {
  eventIn SFBool view1IsActive
  eventOut SFVec3f settextcolor1
  url "vrmlscript:
  function view1IsActive(active) {
    settextcolor1[0] = -33.09;
    settextcolor1[1] = 13.09;
    settextcolor1[2] = 1345;
  }"
```

```
}
ROUTE Birdseye.isBound TO touchScript1.view1IsActive
ROUTE Free.isBound TO touchScript1.view1IsActive
ROUTE touchScript1.settextcolor1 TO adminpath.set_translation

DEF S Script {
  eventIn  SFTime bindTime
  eventOut SFTime startTime
  url "vrmlscript:
  function bindTime(t) {
    startTime = t;
  }"
}
ROUTE Walkthrough.bindTime TO S.bindTime
ROUTE S.startTime TO Walkthrough-TIMER.startTime
#===== end scripts definitions =====#
```

A.4 task2 definitions

```
# task2 definitions
#===== viewpoints definitions =====#
DEF Free Viewpoint {
   description "Free"
   position -43.61 -6.213 1320
}

DEF Birdseye3_narrow Viewpoint {
  description "Birdseye3_narrow"
  position -184 255.7 818.9
  orientation -0.1017 -0.9382 -0.3307 -3.709
  fieldOfView 0.2
}
DEF Birdseye3_wide Viewpoint {
  description "Birdseye3_wide"
  position -184 255.7 918.9
  orientation -0.1017 -0.9382 -0.3307 -3.709
}
DEF Birdseye2_narrow Viewpoint {
  description "Birdseye2_narrow"
  position -223.4 283.7 1235
  orientation 0.5288 0.7682 0.3609 -1.328
  fieldOfView 0.2
}
DEF Birdseye2_wide Viewpoint {
  description "Birdseye2_wide"
  position -223.4 283.7 1335
  orientation 0.5288 0.7682 0.3609 -1.328
}
DEF Birdseye1_narrow Viewpoint {
  description "Birdseye1_narrow"
  position -20.27 249.3 1488
  orientation 0.9996 0.0265 -0.01391 -0.5971
  fieldOfView 0.2
}
DEF Birdseye1_wide Viewpoint {
  description "Birdseye1_wide"
  position -20.27 249.3 1588
  orientation 0.9996 0.0265 -0.01391 -0.5971
}
#===== end viewpoints definitions =====#
```

```
#===== there are no walkthroughs definitions in this task ====#

#===== world spesific objects definitions =====#
DEF V2Grp01 Transform {
 translation 0 0 0
 rotation -1 0 0 -1.571
 children [
  DEF V2Grp01-TIMER TimeSensor {
   loop TRUE cycleInterval 3.333
  }
  DEF Box01 Transform {
   translation 12.09 1091 0
   children [
    DEF Box03_FACES01 Transform {
     translation 2 0 0
     children [
      Shape {
       appearance Appearance {
texture ImageTexture {
 url "container.jpg"
        }
       }
       geometry DEF Box03_FACES01-FACES IndexedFaceSet {
ccw TRUE
solid TRUE
coord DEF Box03_FACES01-COORD Coordinate {
 point [
-4.875 -0.6992 0, -4.875 9.389 0,
1.249 -0.6992 0, 1.249 9.389 0,
1.249 7.188 4.727, -4.875 7.188 4.727,
-4.875 -0.6992 4.727, 1.249 -0.6992 4.727]
       }
texCoord DEF Box03_FACES01-TEXCOORD TextureCoordinate {
 point [
  0.1183 0.02442, 0.1844 0.9104, 0.4596 0.02442,
  0.4589 0.9104, 0.4589 0.6359, 0.1844 0.6359,
  0.1183 0.3657, 0.4596 0.3657, 0.001042 1.003,
  1.001 1.003, 0.6342 0.008983, 0.001042 0.00263,
  0.6785 0.6345, 0.6785 0.95, 0.3631 0.6345,
  0.3631 0.95]
       }
coordIndex [
  0, 1, 2, -1, 3, 2, 1, -1, 3, 1, 4, -1, 5, 4, 1, -1,
  0, 2, 6, -1, 7, 6, 2, -1, 1, 0, 5, -1, 6, 5, 0, -1]
```

```
texCoordIndex [
  12, 13, 14, -1, 15, 14, 13, -1, 3, 1, 4, -1,
  5, 4, 1, -1, 0, 2, 6, -1, 7, 6, 2, -1, 9, 8, 10, -1,
  11, 10, 8, -1]
      }
     }
     ]
    }
   ]
 }
 ]
}
#===== end world spesific objects definitions =====#

#===== infotext definition =====#
string [
"     Task2"
""
"Moet meg ved"
"bygget bak den"
"groenne"
"containeren."
       ]
#===== end infotext definition =====#

#===== there are no scripts definitions in this task =====#
```

A.5 task3 definitions

```
# task3 definitions
#===== viewpoints definitions =====#
DEF Free Viewpoint {
   description "Free"
   position -43.61 -6.213 1320
}

DEF Groundpath Viewpoint {
   description "Groundpath"
   position -34.61 -6.213 1315
}

DEF Walkthrough Viewpoint {
  position -32.61 -4.213 1300
  description "Walkthrough"
}

DEF Birdseye Viewpoint {
  position 187.7 370.6 1500
  orientation 0.7928 -0.5745 -0.2035 -0.9861
  description "Birdseye"
}
#===== end viewpoints definitions =====#

#===== walkthroughs definitions =====#
DEF Walkthrough-TIMER TimeSensor {
loop FALSE
cycleInterval 15 },
DEF Walkthrough-POS-INTERP PositionInterpolator {
  key [0.1, 0.2, 0.3, 0.4, 0.5 0.9, 1 ]
  keyValue [-43.61 -6.692 1310, -34.81 -6.692 1292,
    -1.516 -6.692 1292, -1.516 -6.692 1276,
    -1.516 -2.043 1249, -1.516 -2.043 1056,
    23.77 -0.242 1042 ]
}
DEF Walkthrough-ROT-INTERP OrientationInterpolator {
  key [0.1, 0.2, 0.3, 0.4, 0.5 0.9, 1 ]
  keyValue [0 -1 0 0, 0 -1 0 1.57, 0 -1 0 0,
    0 -1 0 0, 0 -1 0 0, 0 -1 0 0, 0 -1 0 1.58 ]
}
```

```
ROUTE Walkthrough-TIMER.fraction_changed TO
Walkthrough-POS-INTERP.set_fraction
ROUTE Walkthrough-POS-INTERP.value_changed TO
Walkthrough.set_position
ROUTE Walkthrough-TIMER.fraction_changed TO
Walkthrough-ROT-INTERP.set_fraction
ROUTE Walkthrough-ROT-INTERP.value_changed TO
Walkthrough.set_orientation
#===== end walkthroughs definitions =====#

#===== infotext definition =====#
string [
"     Task 2"
""
"Moet opp paa fore-"
"lesning i A blokka"
"Paa Eilert Sundts"
"hus - SV"
       ]
#===== end infotext definition =====#

#===== world spesific objects definitions =====#
DEF SVpath Transform {
  translation -33.09 13.21 1210
  rotation -0.001156 -1 -0.002909 -1.563
  scale 1.414 1.414 1.414
  scaleOrientation -0.008625 -0.02151 0.9997 -0.7856
  children [
    Shape {
      appearance Appearance {
        material Material {
  transparency 0.2 emissiveColor 0.8 0.8 0
}
      }
      geometry DEF SVpath-FACES IndexedFaceSet {
        ccw TRUE
        solid TRUE
        coord DEF SVpath-COORD Coordinate { point [
          116.3 3.301 34.28, 112.2 3.322 35,
  118.1 3.285 33.77, 121.7 3.268 33.91,
  117.8 3.267 43.24, -69.68 -0.09548 -1.795,
          -69.64 -0.08686 0.3851, -58.98 -0.1214 0.3178,
  -56.89 -0.1381 -1.878, 108.7 2.187 22.59,
  110.8 2.171 20.32, -56.76 -0.1736 21.67,
```

```
          -58.78 -0.2011 23.88, -45.85 -0.2168 21.58,
  -45.86 -0.2099 23.78, -41.33 1.559 21.55,
  -41.34 1.566 23.75, -37.01 1.573 23.72,
          -37.03 1.567 21.51, -29.61 2.866 23.66,
  -29.65 2.859 21.45, 115.8 2.124 30.71,
  112.6 2.145 30.53]
        }
        coordIndex [
          2, 0, 3, -1, 1, 3, 0, -1, 3, 1, 4, -1,
  0, 2, 21, -1, 5, 7, 8, -1, 8, 7, 12, -1,
  10, 9, 22, -1, 6, 7, 5, -1, 12, 14, 11, -1,
  8, 12, 11, -1, 20, 9, 10, -1, 14, 16, 13, -1,
  11, 14, 13, -1, 16, 17, 15, -1, 13, 16, 15, -1,
  17, 18, 15, -1, 17, 19, 18, -1, 19, 20, 18, -1,
  19, 9, 20, -1, 0, 21, 22, -1, 22, 21, 10, -1]
        }
    }
  ]
}
#===== end world spesific objects definitions =====#

#===== scripts definitions =====#
DEF touchScript Script {
  eventIn SFBool view2IsActive
  eventOut SFVec3f settextcolor

  url "vrmlscript:
  function view2IsActive(active) {
    settextcolor[0] = -33.09;
    settextcolor[1] = -8.5;
    settextcolor[2] = 1210;
  }"
}
ROUTE Groundpath.isBound TO
touchScript.view2IsActive
ROUTE touchScript.settextcolor TO
SVpath.set_translation

DEF touchScript1 Script {
  eventIn SFBool view1IsActive
  eventOut SFVec3f settextcolor1

  url "vrmlscript:
  function view1IsActive(active) {
```

```
    settextcolor1[0] = -33.09;
    settextcolor1[1] = 13.09;
    settextcolor1[2] = 1210;
  }"
}
ROUTE Birdseye.isBound TO touchScript1.view1IsActive
ROUTE Free.isBound TO touchScript1.view1IsActive
ROUTE touchScript1.settextcolor1 TO SVpath.set_translation
DEF S Script {
  eventIn  SFTime bindTime
  eventOut SFTime startTime
  url "vrmlscript:
  function bindTime(t) {
    startTime = t;
  }"
}
ROUTE Walkthrough.bindTime TO S.bindTime
ROUTE S.startTime TO Walkthrough-TIMER.startTime
#===== end scripts definitions =====#
```

Appendix B

Wireless bandwidth

Network	*Realistic bandwidth (Kbps)*	*Time to download 3D world (8000Kb)*
GSM	9,6	14 minutes
GPRS	44	3 minutes
UMTS	100	1 min, 20 sec
BlueTooth	1000	8 sec
WLAN	6000	1 sec

Figure B.1: Table showing download times for the 3D world on different wireless networks using realistic bandwidth values.

Obtained information from:
http://www.wirelessclueless.com/data.asp
http://www.networkcomputing.com/1202/1202colmolta.html
http://www.troysystems.com/wireless/downloads/books/whitepapers/bluetooth-80211b.pdf

GSM Non-Transparent Circuit Switched Data:
Rhetoric: Highest Theoretical Throughput: 9.6Kbps
Reality: Estimated Actual Throughput: 9.6Kbps

Reliable error free data throughput at 9.6Kbps used by the majority of wireless data applications today including WAP. 9.6Kbps is the highest actual and theoretical throughput for this type of circuit switched data.

GSM Transparent Circuit Switched Data:
Rhetoric: Highest Theoretical Throughput: In excess 40Kbps
Reality: Estimated Actual Throughput: 9.6Kbps

Realistically it is unlikely a user will get a throughput of more that 9.6Kbps using wireless data in this way. Theoretically speeds in excess of 40Kbps have been publicised but these demonstrations typically use very simple text based data that is transferred in non-hostile conditions (i.e. full coverage area) using compression technology. (Obviously no appreciable gain can be made if data is already compressed - which in most cases it will be).

HSCSD (High Speed Circuit Switched Data):
Rhetoric: Highest Theoretical Throughput: 57.6Kbps
Reality: Estimated Actual Throughput: up to 28.8Kbps

Highest theoretical throughput is 57.6Kbps (4 x 14.4Kbps timeslots) - however reality dictates that most networks do not have the capacity or inclination to offer service with more than 3 timeslots (28.8Kbps) and currently terminals are not able to support more than this. Other issues surrounding this technology include the lack of available handsets (currently only Nokia seem genuinely committed to supporting it) and the difficulty “handing over” calls from one cell to another (particularly in congested areas) making it impossible to guarantee a service level in terms of data throughput.

With regard to HSCSD we feel it is unlikely that the speed will increase up to the maximum of four timeslots simultaneously due to the lack of available resources within GSM networks today (something that is unlikely to improve) and the reticence of terminal manufacturers to support this service. In addition we believe that it will be so difficult for operators to provide any kind of guarantee with regard to a quality of service (specific bandwidth) that we stick by our original figures of around 28Kbps for HSCSD.

GPRS (General Packet Radio Service):
Rhetoric: Highest Theoretical Throughput: 171.2Kbps
Reality: Estimated Actual Throughput: up to 44Kbps

EDGE (Enhanced Data Rates for Global Evolution):
Rhetoric: Highest Theoretical Throughput: 384Kbps
Reality: Estimated Actual Throughput: up to 70Kbps

UMTS (3G) (Universal Mobile TelephoneSystem):
Rhetoric: Highest Theoretical Throughput: 384Kbps to 2MB
Reality: Estimated Actual Throughput: up to 100Kbps

It is more difficult to give a specific theoretical throughput for UMTS due to the fact that several “mobility standards” are defined by The International Telecommunications Union (ITU).

The theoretical data rates for each of the “mobility standards” are defined below:
HIGH MOBILITY
144 kbps for rural outdoor mobile use - this data rate is available for environments in which the 3G user is travelling more than 120 kilometres per hour in outdoor environments.
FULL MOBILITY
384 kbps for pedestrian users travelling less than 120 kilometres per hour in urban outdoor environments.
LIMITED MOBILITY
Up to 2 Mbps with low mobility (less than 10 kilometres per hour) in stationary indoor and short range outdoor environments These kinds of maximum data rates that are often talked about when illustrating the potential for 3G technology will only therefore be available in stationary indoor environments.

WLAN (IEEE 802.11b):

Throughputs of 11 Mbps, but realistic values around 6 Mbps because of noice etc. Theoretical range of about 50 meters inhouse, but realistic about 15-20m.

Bluetoth:

10Mbps. The Bluetooth version 2 should reach 10 Mbps.

Appendix C

Interview guide

```
-----------------------------------------------------------------
|  This interview guide is written in norwegian:
|  (muntlig tale)
-----------------------------------------------------------------
```

Hva er ditt overordnede inntrykk av systemet:

Var det nyttig som et navigasjonsverktøy?

Hilket viewpoint foretrakk du?

Tror du at systemet kan hjelpe uerfarne såvel som erfarne Blindern studenter?

Hva syns du bør legges til i systemet? (funksjonalitet osv.)

Kan du tenke deg noen tilfeller der 3D kart vil ha en "fordel" over 2D kart?

Følte du at systemet stjal mye av din oppmerksomhet?

Hvordan syns du det gikk å svitsje oppmerksomheten mellom systemet og RL?

Klarte du å benytte systemet samtidig som du beveget/ orienterte deg rundt på blindern?

Hvordan brukte du systemet? (Bruksmønster)

Følte du at du kunne gå samtidig som du brukte systemet?

Tror du øvelse er en viktig faktor?

Var interfacen lett å bruke?

Prioriter disse argumentene fra 1 til 5 ettersom hvor sann du mener argumentet er. (5 mest enig, 1 minst enig)

Systemet reagerer for langsomt
Oppløsningen (bildekvaliteten er for dårlig)
Det er vanskelig å bruke pennen
Bevegelsene på skjermen er for hakkete
Skjermen er for liten

Hvilken av navigasjons-"aidene" hadde du mest bruk for?
(kompass, groundpath, walktrough, birdseye)

Hvordan brukte du hjelpemidlene?
kompass:

Groundpath:

Walkthrough: