A 3D interface for synchronous collaboration in distributed augmented reality environments

Master thesis Computing Science

University of Groningen
December 19, 2009

Author:	Pieter Bruining (pieter@ustwaddethus.nl)
Supervisors:	Dr. Tobias Isenberg
	Dr. Michael Wilkinson

Version:	1.2
Date:	December 19, 2009

Abstract

This thesis designs a new kind of interface for an augmented reality. Augmented reality is currently not commonly used, but using it can have many advantages. It can, for example, overlay ground cables while digging. The crane operator would be able to see the cables when they are buried. Another example is overlaying an MRI scan during an operation. The surgeons would have more detailed information on where to operate. Also in the field of architecture, a building can be shown using augmented reality which gives the architects or observers a good understanding of what the building is going to look like. The interface designed in this thesis focuses on collaboration. Collaboration between, for example, the architects showing their building to an observer (client). With this interface the architects would be able to adapt the building to the wishes of the client while discussing it.

The interface exists in a 3D environment which is placed in the real world using augmented reality. To make augmented reality more available for common use, a limit is set on the hardware. The hardware that is required must be inexpensive, this places serious limitations on the hardware selection. The interface is focused on collaboration between its users which means interaction is required. However, in order to have interaction an input device is required. 3D input devices often require expensive hardware which must be omitted. Therefore, the hand of the user is used to form an input device in order to realize the interaction. The interface is designed in such a way that users do not have to learn explicitly how to operate the interface because people in general are not expected to have knowledge of 3D interfaces. The properties of the interface are tested with several experiments and the results look promising.

Acknowledgements

In the process of building and writing many people supported me. I would like to thank some people in particular for their help.

First of all I would like to thank my supervisor Tobias Isenberg. Because of his criticism the project started to move forward. During the writing Tobias gave unmissable feedback which helped me finishing it.

Gijs Boer proofread this thesis. His feedback took my thesis to the next level. I would like to thank Gijs for his hours of work.

Kinderdagverblijf Us Twadde Thús, for lending a part of their staff to perform the experiments. With their resources the experiments could be completed in a relative short time.

Alina van der Heide, for her moral support and making sure my focus was on the right thing.

List of Figures

List of Tables

Contents

1 Introduction

Today many applications use AR (Augmented Reality) to combine a virtual world with the real world. In many of these applications this combination is made to better understand the objects in the virtual environment. The advantage is that users can look at and around a virtual object much like a real object. Currently most of these applications require expensive specialized hardware for input and output. Because of this expensive hardware, these AR applications are not available for common use. To make this kind of application available for small businesses and consumers, alternatives for in- and output have to be researched.

The current applications that use AR are very divers', AR is used in many fields. For example, AR is used [19] to train maintenance personnel of airplanes. Another example is ModSAF [10] [28] which is a military simulation system. AR is also used in the gaming sector, Piekarski et al. [29] use AR to play the game Quake. In TV broadcasting, AR is also used, the FoxTrax system [8] highlights the location of the often difficult to see hockey puck as it moves during the game. In the TV broadcasting field, AR is also used to display advertising during sports games. There are many more areas in which AR is used but these examples illustrate the diversity of how AR is used in applications.

AR could help to simplify tasks, for example when creating 3D objects. Such objects are created on a daily basis by 3D modelers (for example architects, industrial designers and game developers). Currently, these people use a classical desktop computer to create their objects. Using AR in this situation could help improve productivity of 3D modelers, because modelers can see real depth with AR. Also the way the objects are displayed (almost like real objects) could help to get a better visual understanding of the object. These benefits, mixed with the feature of working collaboratively on a model, can speed up the process.

The interface of an application that aims on common people has to be easy to learn and understand. In the current AR applications much work is done to enable the AR itself. The interaction part of the application seems to get less attention. In order to have an application for common people, the interface has to have more attention. The first problem to overcome is the way input is received. A way of receiving input has to be defined that is intuitive and does not require expensive hardware. In this situation the hands of the user can be used. Using the hands of the user seems intuitive because people use their hands to interact with real objects every day.

Networking is important in applications. Networking can be used to work together on the same project or play a game with or against each other. With AR applications networking can be used to work collaboratively in the same virtual environment. Working together can increase the productivity. Users can, for example, explain things to each other while physically at different locations. A problem with concurrent users in one virtual environment is the understandability. Users have to have a way to understand what is happening to not get confused. The interface of such an application has to cope with this.

All these aspects mentioned are not combined in one application. To fill this gap in the field of AR, a project is initiated. The goal of this project is to create a prototype application that realizes a virtual environment placed in the real world. Multiple concurrent users should be able to interact with this virtual environment by purely using their hands. With this application users would be able to work collaboratively in the same environment. To create such an application a real world case is defined which is described in the following paragraph.

1.1 The Digital Maquette case

When an architect designs a building and the customer wants to see what the end result will look like, usually a maquette (a scale model) is built. This maquette is used to give the customer a visual impression of how the building is going to be. People can walk around the maquette, look inside the building and they can see what the building would look like in reality. However, to create such a maquette, time and resources are required. Building maquettes can take weeks to complete. A drawback of a real maquette is that it cannot easily be altered after it is created. If this maquette would be a virtual maquette by using, for example, an HMD (Head Mounted Display) and AR, the visual appearance would be much the same as the real thing. Unlike a real maquette, this virtual model could be altered anytime and usually at a much faster rate. With the use of AR, structural adjustments to the model only need re-rendering instead of rebuilding the physical maquette. Different colors could also be tried out, to see which colors fit best requiring little effort. Building the maquette could be done with multiple users who do not have to be at the same location. It would save time, resources and money and it offers more possibilities than physical maquettes. The main drawback of a virtual maquettes is that they cannot be touched like physical maquettes.

A difference between using AR instead of VR (Virtual Reality), is that AR can be used in any office or room because with AR the surroundings are visible. VR would require a room where there are no obstacles present so that the user does not bumps into something while walking around. In this case, a table can be used to display the model on a fixed location and give the user the ability to walk around the table to see the model from different angles.

Because the digital maquette is a virtual environment built from virtual objects, this environment can be replicated to different locations. This gives the architect the possibility to discuss the preliminary version of a design with the customer who could be at a different location. However, this creates a gap in the communication between the customer and architect. The customer or architect now need a way to let the other know what he or she is talking about since they cannot physically see each other. They need a way to pinpoint or select parts of the model. Nowadays, this could be done by using the keyboard or mouse. However, since the users can walk around freely, a keyboard or mouse are quickly ruled out as a possible input device. Therefore, an alternative way of input has to be defined.

The prototype of this case is named the Digital Maquette system. In the project four different research fields are distinguished, hand tracking, hand pose estimation, replication and interfacing. The hand tracking research field searches for a solution to find the hand in a video feed. In order to use the hand as input, the pose of the hand must also be known. The hand pose estimation research field searches for a solution on the pose problem. In order to display the same virtual environment on different locations, the replication research field is defined. The last research field is the interface field, this field focusses on creating an interface for the Digital Maquette system. For each research field a different master thesis is set up. M. Fremouw dedicated his master thesis [15] on the hand tracking field. G. Boer is researching [4] the hand pose estimation. The replication field is researched [25] by H. Lenting. For the last research field, interfacing, this thesis is defined. In the following paragraph the challenges found in the research field "interacting" are described.

1.2 Challenges

The main challenge is how to create an intuitive 3D interface for AR with just the users' hands as input. For this research it is assumed that the position of the thumb and the index finger of a hand can be tracked. This means that the interface has to deal with 3D finger input. The goal is to create an interface which can easily be used by people who are not familiar with it. The interface has to be in the augmented world, this means that the users have to be able interact with the objects in the augmented world and not with a remote control system [17]. Because multiple concurrent users can work together, the interface must work in such a way that, when a user changes something, the other users should be able to understand what is happening. With this interface the following tasks have to be possible:

- point out a virtual object to remote users,
- move, rotate and scale a virtual object,
- add a new predefined virtual object to the augmented world, and
- remove a virtual object from the augmented world.

In the following section a summary of this research is given. In this summary the interface that supports these tasks is described briefly.

1.3 Results

This section gives a small preview of the result made in this thesis. The interface designed in this thesis used real-life collaboration concepts. In real-life, people see each other and, therefore, can understand the actions of each other. To make it possible that users can understand each other, the designed interface shows the hand of a user to other users. Actions of a user can, therefore, be seen by other users. According to the experiments this method works well, users indicate that they understand what other users are doing.

This thesis makes an effort to create an intuitive interface. In the experiments the time required for the users to learn how to use the interface is used as a measurement to approach the intuitiveness. On average the users needed less then half of the expected time to complete the defined task. Therefore, the interface performs reasonable in terms of intuitiveness.

In Chapter 3, Concept, the interface is explained in detail. In the next section the organization is given for this thesis.

1.4 Organization

This thesis is organized in the following way:

Chapter 2, Related Work, describes the current and prior research related to this thesis. It explains existing systems and techniques that can be used by this research.

Chapter 3, Concept, defines a 3D concurrent interface for the Digital Maquette case. Furthermore, the chapter tries to connect all required techniques in such a way that a prototype can be built.

Chapter 4, Realization, describes the implementation of the concept defined in Chapter 3. Furthermore, encountered problems are described in order to make other developers aware of these problems.

Chapter 5, Experiments, is a chapter which defines experiments to evaluate the 3D concurrent interface defined and realized in Chapters 3 and 4, respectively.

Chapter 6, Results, describes all the data acquired from the experiments defined in Chapter 5. At the end of this chapter a discussion is created about results that stand out.

Chapter 7, Conclusion, concludes and summarizes this thesis. Furthermore, it describes directions for future work.

2 Related Work

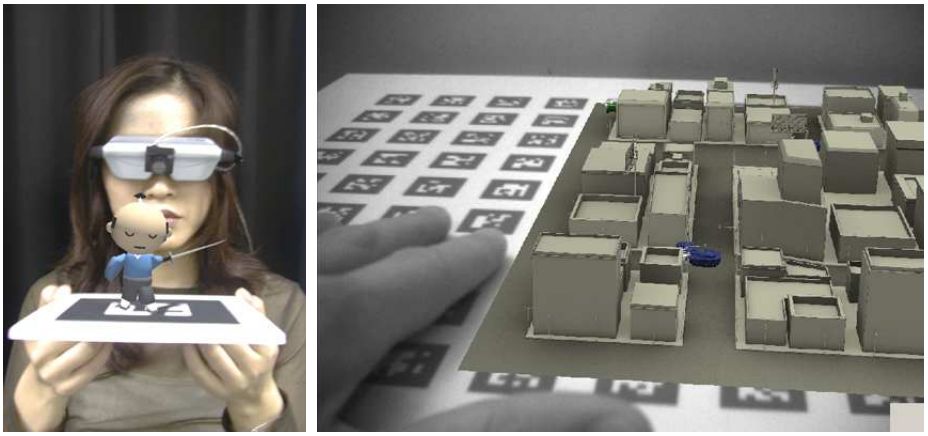
For a collaborative 3D user interface that uses AR, a number of different systems are required. In the field of AR much research is already available, therefore part of the required systems are already available. In this chapter an overview is given of the related work and prior art. The concepts given in this chapter can be used as basis for this research. The first system that is required is a system that enables the use of AR, in the first paragraph such systems are listed.

2.1 Augmented Reality

In the field of computer science, AR combines VR with the real world. Combining the two worlds can be done by blending the rendered objects of the VR into a video feed of the real world. The main advantage of using AR instead of VR, is that the user can walk around freely. In the Digital Maquette case this is important because otherwise the users cannot see the other side of a maquette. In order to blend the VR objects in the video feed, a reference between these two worlds needs to exist. Marker systems are systems that can realize this kind of reference. ARToolkit [35], ARTag [14], and ARToolkitPlus [22] are such marker systems. Figure 1a is a picture of a user holding a marker, this marker is extracted from the video feed that is captured by the camera on the users head. The position and orientation is calculated from the data using the marker. When the position and orientation is known, a virtual object can be placed relative to the marker, for example on top of the marker. The other marker systems work in the same manner, in Figure 1b a more complex example is given. This example is a tank game based on the marker system ARTag. The ARTag library has the feature to use multiple markers to calculate one big square, for example a table. This can be useful to minimize possible error, but this is more expensive in terms of calculation power. For the Digital Maquette case all three systems are suitable, all three systems work roughly the same.

2.2 Glove-based systems

When having a reference between the real world and the virtual world, the problem of placing virtual objects has been solved, however, this is not enough to create an interactive system. To realize an interactive system an input device is needed, this input device, however, cannot be a classical keyboard and or mouse. A user has to be able to walk around the virtual world and still be able to interact with the objects. Thomas and Piekarski [32] use a glove-based input device to create a 3D interface for the user. The gloves are used to control a menu, a user can access the menu by pinching in the glove. The



(a) ARToolKit, from [24] (b) ARTag - Tank game, from [9]

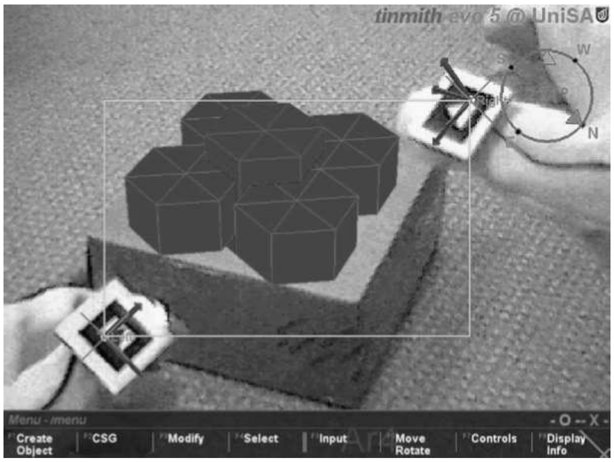
Figure 1: Marker systems

menu is displayed at the bottom of the images in Figure 2. They also describe three general selection techniques: two-handed framing, line of sight and laser beam. These selection techniques can be used to select virtual objects and interact with them and can be accessed by the menu. Two-handed framing is a technique which is similar when selecting with a classical 2D mouse, in for example a file manager. The first point of the selection box is the thumb of the first hand and the second point of the selection box is the thumb of the other hand, this is illustrated in Figure 2a. The result of this selection technique is that the virtual object in the selection box is selected for interaction.

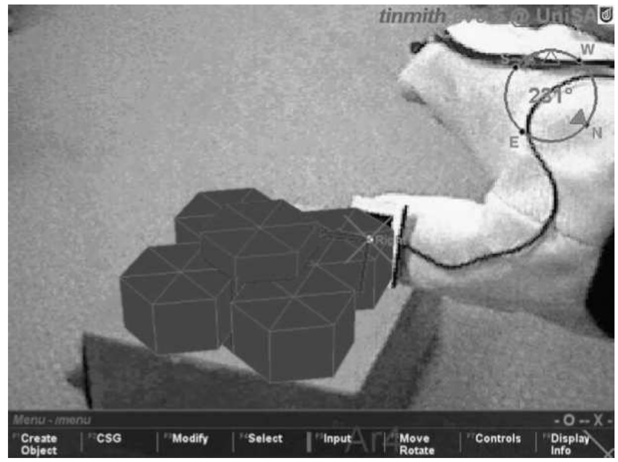
With the line of sight technique the user can place his/her thumb behind or in front of the object. The technique then selects the object in the line of sight, from the camera to the thumb. In Figure 2b an illustration is given. It must be noted that the virtual objects are always drawn on top of the video feed. This has the effect that the virtual objects are always in front of the hand, even when the virtual object is further away than the hand with respect to camera.

The last selection technique is the laser beam technique, the thumb is extended with a beam, as displayed in Figure 2c. With this beam the user is able to select a virtual object that is far away only by pointing his thumb at the object. This technique can increase productivity in large virtual environments because the user does not have to walk to the virtual objects in order to interact with it. In the Digital Maquette case it probably is not necessary to select objects from a distance because almost all objects are within the range of the users' hand.

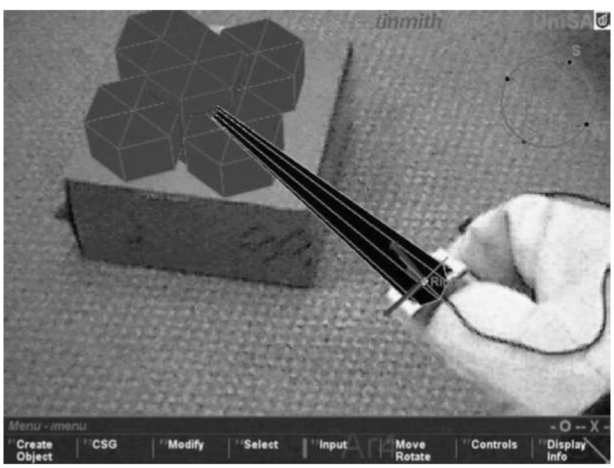
Thomas and Piekarski also have a method of inputting characters, the glove-based keyboard. This keyboard is illustrated in Figure 2d. Inputting characters can be used in the Digital Maquette case for labeling objects. This is not a direct requirement of the case, but labeling is something that might be interesting. The characters are set on a grid and the user can input them by pressing one or two fingers to input one character. This keyboard is limited to 49 characters because a big character set expands the grid on which the characters are placed. The input complexity increases when the grid is bigger, therefore it is important to keep the grid as small as possible. This way of inputting would require physical buttons on the fingertips of the gloves (described in [32]) and impact the ability for users to perform other tasks, because they may accidentally be activated.



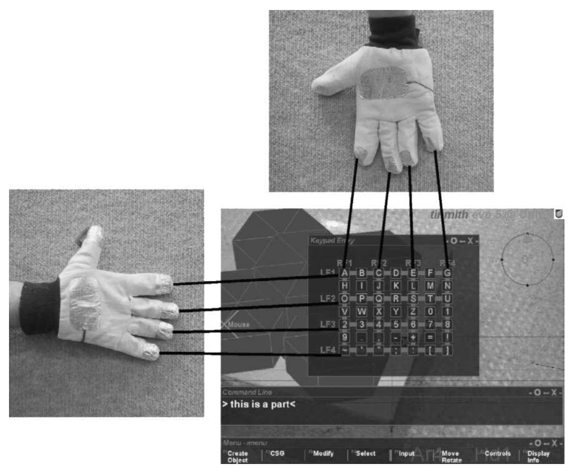
(a) Two-handed framing



(b) Line of sight



(c) Laser beam



(d) Keyboard

Figure 2: Glove-based input techniques, from [32]

Another pinch glove menu system is that of Bowman et al. [7]. This menu system, TULIP, is attached to the fingers of the user as displayed in Figure 3. The user can activate an item by pinching the connected finger. With this approach, the full position and orientation of the hand is needed in order to display and navigate the menu correctly. This menu system can be used in the Digital Maquette case to add or remove object from the virtual environment. Bowman et al. also describes a way of entering text with the pinch gloves, this method uses a QWERTY layout which is displayed on screen. The user can move the hand to select a row from the virtual keyboard. When the row is selected the key can be entered by pinching the corresponding finger. This method leaves the inner keys such as ‘g’ and ‘h’, which can be selected by rotating the hand inward. The research depicts this motion as awkward and also provided the alternative method of pinching the thumb to both the index and middle fingers to select an inner key.

A general drawback of glove-based systems is the use of gloves, because gloves limit the user. With gloves users cannot do other things while interacting, for example write down notes on paper. This limitation is a serious drawback in the usability with respect to the Digital Maquette case. Bowman et al. [6] also compared the TULIP menu system against other menu systems. The results of the comparison is that users need a little more time to learn the TULIP menu system, but when it is learned it handles just as fast as the other menu systems. The big advantage of the TULIP menu system is that the user has less arm and hand strain then the alternative tested menu systems. In the Digital Maquette case this menu system can be used, but it requires that all fingers are tracked by the hand tracker. A menu system in general can realize adding and removing objects from the virtual environment.

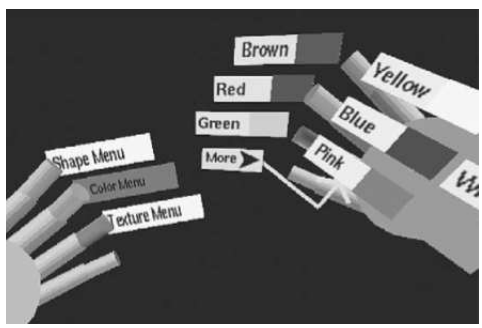
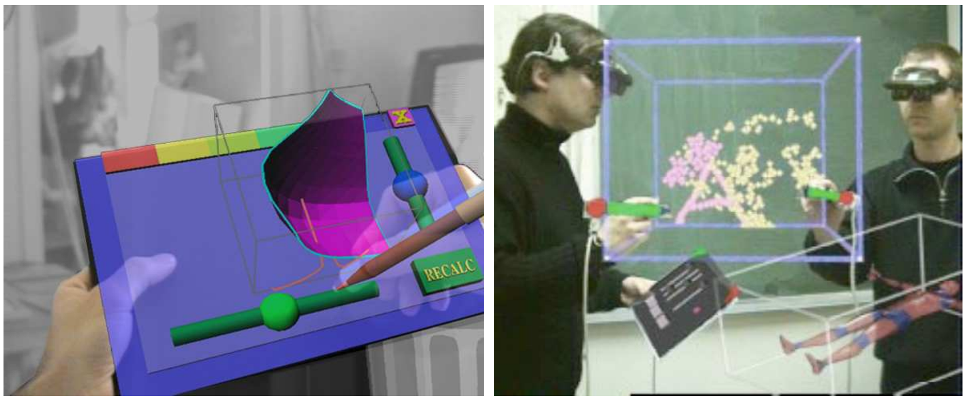


Figure 3: TULIP menu system, from [7]

2.3 Interaction systems

For the Digital Maquette system it is important to have a high usability, because AR systems are not very common which means common people do not have a lot of knowledge of these type of systems. An important part of the usability is the way users are required to interact, therefore additional interactive systems which can be used with AR are reviewed that do not involve gloves as input.

The PIP (Personal Interaction Panel) [39] is an example of a 3D interface for AR. The interface consists of a physical tablet with a physical pen. The tablet and pen are tracked and presented in AR. Because the tablet is known in the augmented space, objects can be mapped on it. With the pen, the objects can be manipulated. This is actually a 2D interface mapped onto 3D space. To illustrate this, the authors use buttons and sliders in the augmented world on the tablet, an example is displayed in Figure 4a. For the PIP system it is required that the users carry a tablet and pen; this is extra hardware which should not be required because extra hardware limits the users. However, the physical tablet and pen give the user tactile feedback. Because of the extra hardware the PIP system is considered not useful with respect to the Digital Maquette case. The idea of a control panel, however can be used.



(a) The PIP Interface, from [31] (b) Multiuser, from [30]

Figure 4: The PIP in action

Schmalstieg et al. [30] use the PIP interface to create a multiuser environment. This environment shows the virtual PIP devices of every user in the augmented world. Which makes the other users aware of what is happening if a user performs certain actions. The awareness aids users to understand what other users are doing [11], but again this system requires additional hardware which makes it not very useful in the Digital Maquette case. The idea of showing the interface of the concurrent user can however, improve usability. Figure 4b illustrates how two concurrent users can work together.

Feiner et al. [13] present a way of creating a window-based interface for AR. They use 2D windows which are displayed in the real world using AR. Figure 5 is an illustration of this window-based system. The interaction with the windows is done by a normal pointing device and a normal keyboard. This would require the users to use a keyboard and mouse, so obviously, this system cannot be used in the Digital Maquette case.

Wilson created the input device XWand [38]. This input device registers movement, orientation and has several buttons. With the XWand, remote operations are easy to perform, but this system requires additional hardware (the XWand itself). This makes it not very useful in the Digital Maquette system because every user has to operate an XWand in order to interact. Another problem with this input device is that it is not a product but a

Figure 5: Window-based interface, from [13]

prototype, and after 2004 the development was halted. In Figure 6 a XWand is displayed.

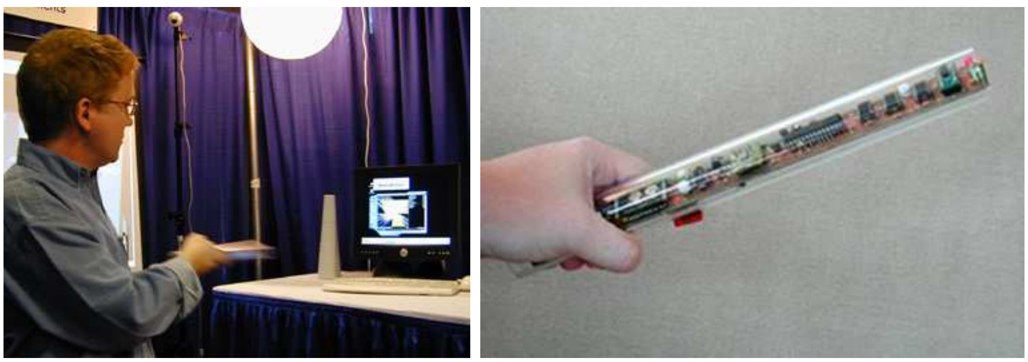
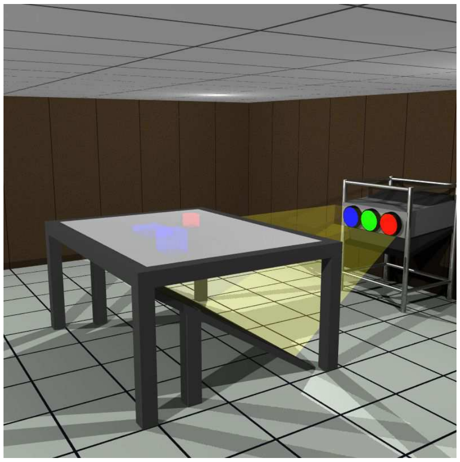
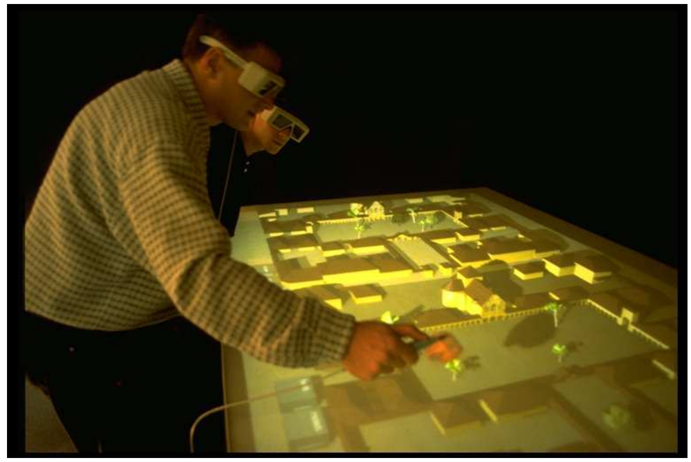


Figure 6: The XWand, from [38]

Another interaction system is The Responsive Workbench [23] created by Krüger et al. The Responsive Workbench uses a projector to show information through a mirror on the bottom side of the table, as displayed in Figure 7a. By using shutter glasses, Figure 7b, this projected image can be experienced by the user as 3D. To display the right point of view for the user a 6 DOF tracker is used to track the users' head. In order to interact with the workbench, a pen or gloves are tracked by the system. The Responsive Workbench system requires an adapted table, a projector, tracking devices for the users' head and tracking devices for the pen or gloves. Because of all these requirements this system is rather expensive and is difficult to set up. For this reason this system is considered not suitable for the Digital Maquette system.



(a) Schematic view



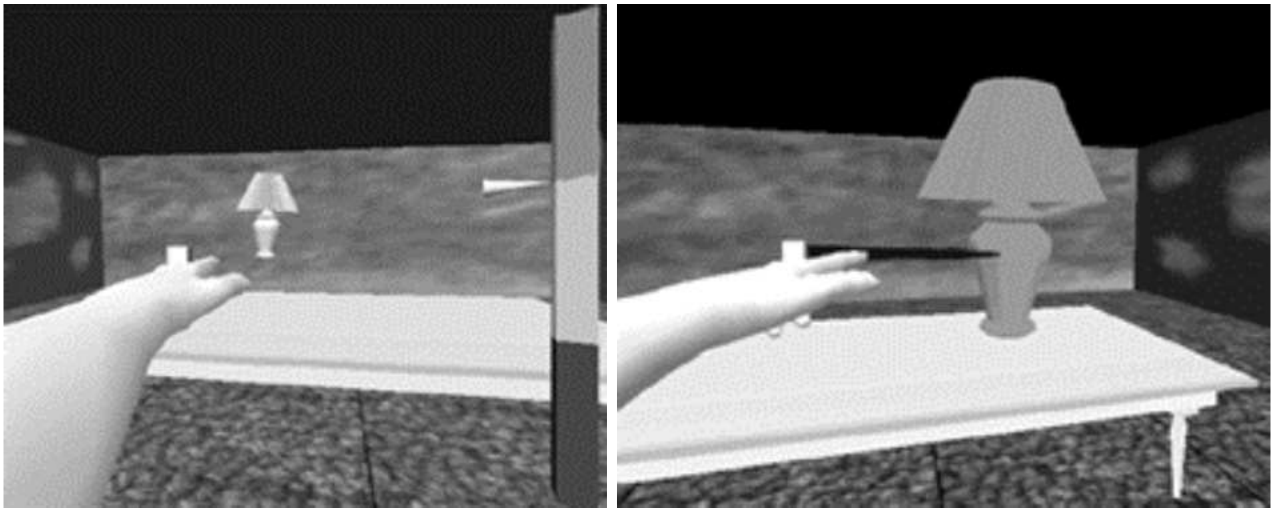
(b) User working with the workbench

Figure 7: The Responsive Workbench, from [3]

2.4 Interaction and interfacing techniques

In order to properly display the AR and to create an optimal interaction system, specialized techniques can be used. This paragraph reviews a number of techniques that can improve the usability. Bowman et al. [5] evaluate a number of techniques for grabbing and manipulation. The Go-Go technique (Figure 8a) is an arm extension technique which lets the user extend his/her arm in the virtual environment to grab an object. With ray-casting (Figure 8b), the user can shoot a ray to an object. When the object is selected with the Go-Go or the ray-casting technique, the user can manipulate the object. The advantage of ray-casting is that the selection is easy, however, rotating the object with ray-casting is only useful over the axis of the ray. The Go-Go technique suffers from imprecise grabbing and it has a finite range. Therefore, the authors propose a new method, the HOMER method. This is a combination of the Go-Go and the ray-casting technique. The user selects the object with a ray and then the hand extends to the object, in this way the user can manipulate the object in a useful way. The ray-casting, Go-Go and HOMER techniques are used to select objects which are further away than the user can reach. With the Digital Maquette case, everything is happening on the table in front of the user, which means that almost everything is within the reach of the users' hand. Therefore, using these selection techniques would only add complexity to the interface and not have any real benefits.

An AR can result in a lot of information for the user, especially when multiple users are active in the augmented space. Julier et al. [21] describe an automated information filtering algorithm. This algorithm filters irrelevant information in such a way that the interface becomes more readable for the user. The example used in the paper is the display of a schematic view of a building on top of the real building, where the unfiltered version overlays the complete structure and the filtered version overlays only the visible part of



(a) Go-Go (b) Raycasting

Figure 8: Grabbing and manipulation, from [5]

the building. Information filtering can be very useful when the interface becomes crowded with information. Because the case is to build a Digital Maquette the objects should not hold a lot of additional information, therefore using this method does not seem necessary.

There are a lot of different interaction techniques, not all techniques fit well in the Digital Maquette case. Chris Hand [18] does a survey of 3D interaction techniques, which are useful in different fields of 3D systems. Gregory et al. [17] uses a haptic display for remote controlling a 3D environment. This haptic display looks like the PIP interface with the difference that this display is not part of the virtual environment while the PIP interface is. Bowman et al. propose new research directions about 3D interaction in [1]. Bowman tries to set new research directions for the field of 3D interfaces because the growth in this area is slowed down since 2000 according to [1]. This paper summarizes many techniques that can be used in combination with AR.

2.5 Summary

In the field of AR many systems are available, these systems often require additional hardware. For example, the PIP system uses a tablet and pen for interaction. For concurrent users, the PIP system would be suitable because users can see each others actions. Another AR system uses a table as projection screen for displaying information. These specialized hardware makes the system expensive and hard to set up.

Many interaction systems are based on glove-based input. Interaction techniques using gloves can, to a certain degree, be used in the Digital Maquette system because the hand of the user is tracked. However, glove-based systems themselves cannot be used because the gloves limits the user in executing other tasks while interacting. Glove-based interaction techniques are often designed for single user usage. The techniques display information only on the display of the user. This means that other concurrent users are not informed with the actions the user executes. In the interface of the Digital Maquette system this can be changed, users can be informed of actions performed by other users. In other words, some interaction techniques based on gloves can be used in the Digital Maquette case.

Interfacing techniques like HOMER, Go-Go and Raycasting are used to select objects that are not within reach of the user for manipulation. The virtual environment in the Digital Maquette case is placed on a table. Users can walk around it and because of this almost every virtual object is within reach. Therefore, these selection techniques are unnecessary for the Digital Maquette system.

3 Concept

While AR is not new, it is not widely available for every-day use. This can be attributed to the price and availability of specialized hardware, such as an HMD. This meant that small businesses and consumers could not afford the hardware. This has changed, over the years electronics became cheaper and the required specialized hardware can now be acquired for less than 500 euros [27, 34], which makes it more affordable for common usage. This thesis tries to create a 3D collaborate interface which runs on inexpensive hardware in order to bring the AR technology to common people.

The 3D collaborate interface in this thesis is focused on the Digital Maquette case. In this chapter, a concept is given that tries to comply with the Digital Maquette case. In order to create a concept of the Digital Maquette system, requirements of the system must be defined. These requirements are stated in the next section. After that, the general setup of the Digital Maquette system is given. This general setup will define the Digital Maquette system on a global level. After the general setup the important specifics of the Digital Maquette system are described. At the end of this chapter a summary is given of this chapter.

3.1 Requirements

In order to design the interface, requirements must be set. In this section the requirements of the Digital Maquette system are defined. In Table 1 the requirements are listed. For each requirement a motivation is given. These requirements are used throughout the concept as a basis for the design decisions. The most important requirements are discussed in the next paragraph.

In the field of AR and VR a number of systems are already available. Most of these systems require expensive hardware. The target area of the Digital Maquette case are small businesses and costumers. This area does not always have the funds to acquire expensive hardware. One of the main requirements for this system is that it should not require expensive hardware so that small businesses and customers can use this system. Using the Digital Maquette system should give an experience close to a physical maquette. Keeping the gap between a physical maquette and a digital maquette small also keeps the transition small. This can help people to switch to a digital maquette faster. With a physical maquette people can freely move around it and pinpoint objects to others at any given location. It is very important that the Digital Maquette system also has this experience in usage. Users should not be limited by input devices attached to their hands. When a user has, for example, a glove on his/her hand the user is limited by the working radius of the glove. With gloves the user is also limited in executing other actions, for example, writing down notes. Therefore, no additional hardware should be required for input. Physical maquettes are often used to discuss a design, for example of a building. While discussing, people communicate with each other by, for example, pointing at certain objects. This kind of communication and collaboration should also be possible in the Digital Maquette system because it is one of the main reasons why a physical maquettes is made. A complete overview of all requirements is shown in Table 1. Every requirement is given a unique number, this number is used as reference in the rest of the concept.

#	Requirement / Motivation
1	**The Digital Maquette should be available for common use and, therefore, not expensive.** There already are expensive systems that can display virtual environments. The key of the Digital Maquette case is that it should be available for consumers and small businesses.
2	**The virtual environment should be placed on a real world table.** The way the Digital Maquette is displayed should be close to the way a physical maquette is displayed and real maquettes are usually placed on a table.
3	**Users must be able to walk around the virtual environment / real world table.** The experience of viewing the Digital Maquette should be close to viewing a physical maquette because in this way the users can adopt the system faster. When a user wants to view the rear of a physical maquette the user can walk around it. Therefore, this should also be possible with the Digital Maquette.
4	**Users should not need additional hardware for input.** Additional hardware can be confusing for the user. With additional hardware users cannot move around freely which is required in requirement 3. Additional hardware introduces additional costs and the complete system should not be expensive according to requirement 1.
5	**Users should be able to work together in the same virtual environment from a different room or buiding and understand each others actions.** The main reason behind building a maquette is showing a design to other people. In order to explain the design to other users, users need to work collaboratively.
6	**Users should be able to add/remove objects to/from the virtual environment.** In order to edit the design adding and removing objects has to be available. This can be used to show different designs.
7	**Users should be able to move/scale/rotate objects in the virtual environment.** Adding and removing objects is not enough to show different designs. Objects also have to be placed at the right location and with the right orientation and scale.
8	**A user should be able to pinpoint a virtual object to another user.** Users can work collaboratively, but this does not mean that the users are in the same room. Therefore the users need to be able to pinpoint objects to each other, to show what their talking about.
9	**The system should be intuitive, users should be able to work with the system without explicit learning.** The Digital Maquette case focusses on architects showing designs to clients. Clients do not necessarily have experience with the Digital Maquette system or with any other 3D interactive system for that matter. Architects do not want to explain the Digital Maquette system to every client, this would be too time-consuming. Therefore, without explicit learning users should be able to use the Digital Maquette system.

Table 1: Requirements of the Digital Maquette case

3.2 General setup

In this section an overview is given of the whole Digital Maquette system. This thesis focusses on the design of the interface of the Digital Maquette system. In order to view and interact with the interface, in- and output methods need to be available. These methods must first be defined because using a different method of in- or output can affect the interface.

To show the Digital Maquette to the user a displaying technique must be chosen. VR might be an option, but using VR would require a room with no obstacles present so that the user would not bump into something. Because users should be able to walk around the maquette (requirement 3). Such a room is expensive to setup, also consumers usually do not have a room to spare. Therefore, when using VR requirement 1 would not be met. Using AR instead of VR would give the users awareness of what is happening around them. Using AR would therefore not require a room without obstacles. This makes AR more suitable for the Digital Maquette system.

Every user in the environment views the system from his/her own point of view. This is required because of requirement 3. Users should be able to walk around the maquette to view, for example, the rear of a virtual building. To realize such an output system, an HMD can be used, but also the Responsive Workbench [23] is a system which can be used. When users have a personal point of view, the interface of the Digital Maquette system is drawn for every user separately. This means that the location and orientation of the head of the user relative to the table where the virtual environment is located must be known. This can be realized by a global tracker, used in [23], or a marker system like [14], [22] and [35].

The Responsive Workbench is an expensive system. Requirement 1 will not be met if this system is used. Therefore, using an HMD seems a better approach. Using an HMD will affect the interface of the Digital Maquette system. Currently, there are no affordable see-through HMD's available like the Lumus Optical HMD [26]. This means that a background video of the real world must be present in the HMD to give the users awareness of the physical objects around them. The interface of the Digital Maquette system must provide such a video stream. This also means that a camera has to be attached to the users' head. Marker systems can use the video feed from the head mounted camera to calculate the position and orientation of the table where the virtual environment should be displayed. This means that no tracking device is required when a camera is present on the users' head. Using a marker system in this situation seems a better approach because it saves additional costs of the tracking devices.

According to requirement 4 and 9 the input method should be intuitive for users to use and not require additional hardware. Using the users' hand for input seems to be an intuitive method because people use hands to interact with real world objects every day. It is assumed that at least the index finger and thumb of one hand of the user can be tracked by the hand tracker [15] and the hand pose estimation [4]. This requires no additional hardware and the users are also not limited in their hand movements by using this kind of input. The drawback of these techniques is that it is research in progress. Currently, the two techniques do not give enough output for the interface to work with. But it is expected that it will work in the near future. The user input section of this chapter defines in more detail why and how this way of input is used.

To let people work collaboratively across rooms or buildings, communication is needed between the user interfaces of each user. This communication can be done over IP networks which makes it compatible with the Internet. Using an already available network can significantly decrease the costs for the end users. This network can distribute all the actions of a user to the other users, making it possible to work together.

The general setup is illustrated in Figure 9. This is a schematic overview of how the components relate to each other.

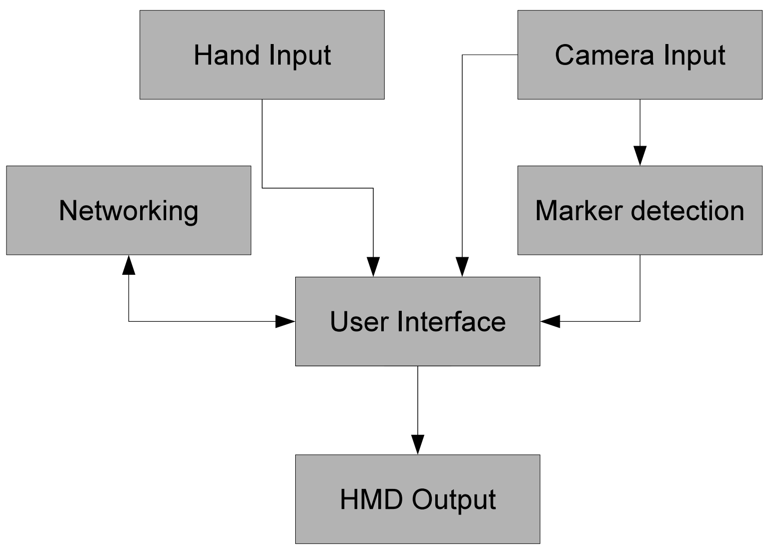


Figure 9: Components of the Digital Maquette system

3.3 User input

The user needs to be able to interact with the system, as described earlier a normal keyboard and mouse are not an option. Specialized 3D interaction devices like the Space Navigator [2] are also not ideal in this situation because these devices require the user to stay at one place while interacting. Requirement 3 specifies that a user must be able to move freely. This means there needs to be an alternative way of input. A possible intuitive way would be using the users' hands for interaction, because humans already have experience interacting with real objects with their hands. Registration of the movement of the hands is a difficult task and outside the scope of this research. Wang et al. [36] have an possible solution for this. For this research it is assumed that the thumb and index finger of one hand can be tracked in 3D space because of the limitations of the hand tracking and hand pose estimation. This assumption/limitation should not limit the interface because with the thumb and index finger it is possible to define enough actions for the Digital Maquette case.

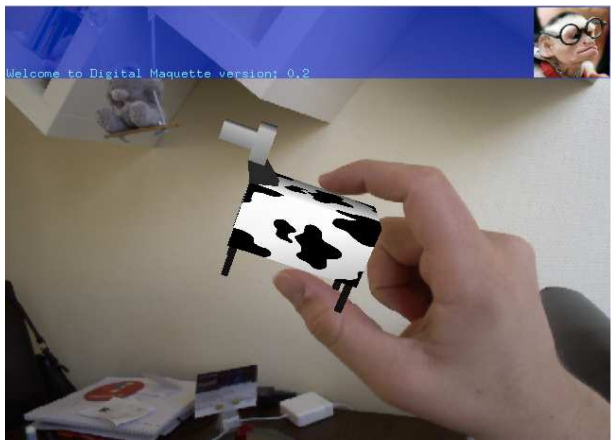
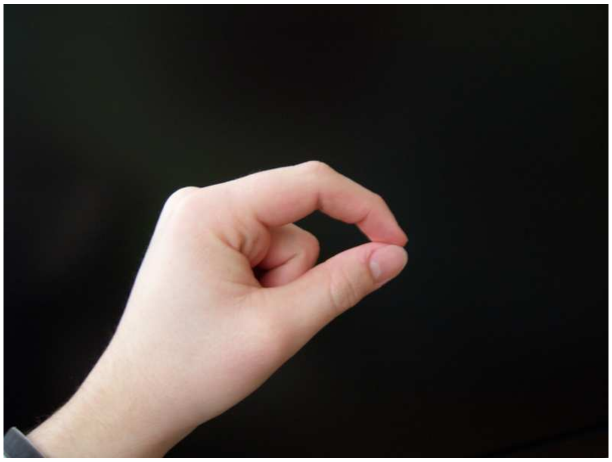


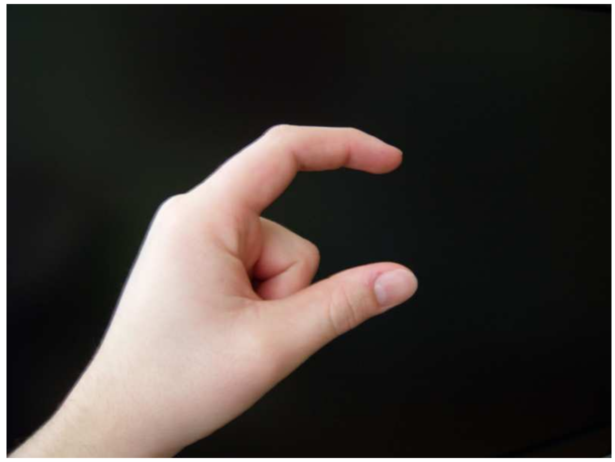
Figure 10: Grabbing a virtual object (mock-up)

In order to establish interaction, actions have to be defined on the hand. Because the thumb and index finger are tracked it would be possible to grab a virtual object, illustrated in Figure 10 which is a mock-up of this technique. This is much like how a normal person would, for example, pick up a pen. With this type of interaction a user would easily grab through the object because a virtual object does not have any feedback (touch). Grabbing through an object can confuse the user. Another problem with this approach is that users cannot grab objects that are larger than the maximum distance between the thumb and index finger. Therefore, this approach might not be an ideal approach.

Another approach is using the pinch technique [33], although this approach is in 2D, it can be translated in 3D because the finger and thumb positions are known in 3D. With this technique it is possible to create multiple actions with only a pinch as input. An illustration of the pinch and release input is given in Figure 11. The input actions are defined in Table 2, these actions can be extended through a menu system. To establish, for example, the action "remove object", the menu can be extended by a simple item remove. Menu systems could be intuitive for most people because they are, to some degree, familiar with traditional computer interfaces which often use the concept of menu systems. A menu system, which can be abstracted to a list of choices, is not only used in computer systems. Therefore, most people are familiar with menu systems. A menu system is used in the Digital Maquette system to add functionality. With the menu system, all the interaction



(a) Pinch



(b) Release

Figure 11: Finger/Thumb input method

can be done through the pinch/release action and hand movement. The effect of having only two basic forms of interaction should result in an easy to learn interface which meets requirement 9.

User action	System reaction
Index finger enters an object	Object becomes semi-transparent
Pinch/release in an object	Menu is displayed in the object and the object stays transparant
Pinch in menu item	Action coupled to the item is executed
Pinch menu move item	Object position is coupled to thumb and index finger position
Release menu move item	Object is decoupled from thumb and finger position

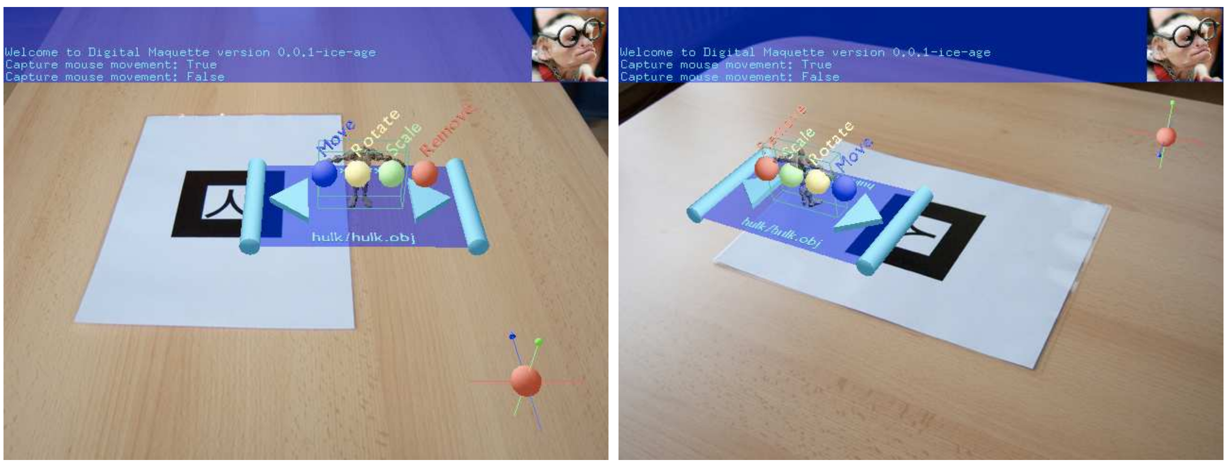
Table 2: Basic interaction: Action / Reaction

3.4 Menu system

In Section 3.3, reasons are given for using a menu system. This section will state the details of the menu component. As defined in Table 2, users have to pinch in an object in order to activate the menu. The menu consists of an arbitrary number of items, each item has a different action, label and color. Figure 12 shows two examples of a menu being displayed. The reason behind the difference in label and color is to maximize the understandability of the menu system. The use of colors aid humans to learn the interface faster [16]. Every color should have a different meaning which is consistent throughout the whole interface.

In this way the user will familiarize faster with the interface. When a user is familiar with the interface, he or she probably does not need to read the label anymore to know what kind of action the item represents.

The menu is defined by a number of items, these items are displayed to the user. An item is a colored sphere with a label attached to it. Instead of the spheres, small 3D models, could also be used which is much like icons in a 2D interface. Small 3D models may look better but it can possibly confuse the user, the user might think that the 3D model belongs to the Digital Maquette. Using only colors simplifies the interface which should have a positive effect on usability. The label should not contain a lot of words, ideally one word per item. With smaller labels, users can read the labels faster and when the right words are used to describe the action behind the item, the user should understand the interface faster.



(a) First angle (b) Second angle

Figure 12: Object menu

To activate an item, the user can pinch in the sphere representing the desired action. Figure 12 displays an active menu for the object attached to the blue panel. The spheres of the menu items are displayed for every concurrent user at exactly the same location. Other users can activate these menu items at the same way as the user that pinched in the object. The text label however, is not displayed at the same location and orientation for every user. The text label of an item is rotated towards the viewing position of the user for which the view is rendered. This makes the labels readable for every user viewing from an arbitrary viewing point, this should aid the user to better understand the interface. Figure 12a shows the same object as Figure 12b from a different angle. Because of the text rotation, the text of the labels are readable from both angles.

A user selects an object when the users' hand is moved within the boundaries of an object. When an object is selected it becomes semi-transparent and a bounding box is displayed around the object. While the object is semi-transparent, the user can see through the object to view objects behind the selected object. The bounding box shows the user exactly where the object is placed in the virtual environment. Therefore, showing the bounding box aid users while moving objects. Because the object is semi-transparant and a bounding box is displayed, the user is informed that the object is selected for interaction. The user can pinch in a selected object to activate the menu for that object.

The standard menu of an object is defined by the following items:

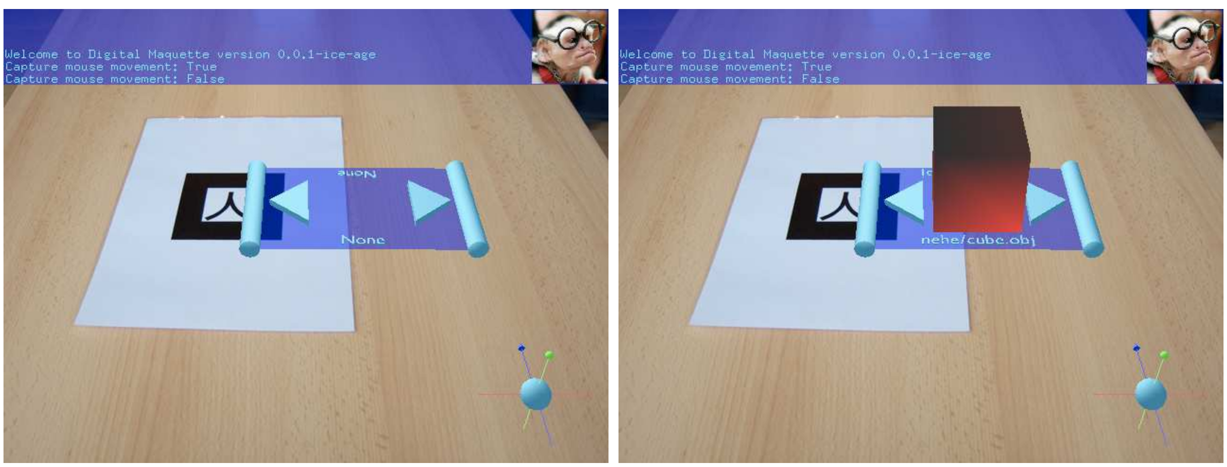
- Move (Blue)
- Rotate (Yellow)
- Scale (Green)
- Remove (Red)

An object can add menu items to fit the needs for that object. Throughout the user interface, every action type has the same consistent color. “Move” is an item that the user can activate by pinching and holding the pinch. The user can move the object around while holding the pinch. When the user releases the pinch while moving, the object will be placed at the last position before the release. While moving an object, the orientation is not changed. This is important because a user might want to move an object and not change the orientation. This would almost be impossible when moving and rotating is combined, because humans tend to change the orientation of their hand while moving it. Rotating is similar to moving, holding the pinch and rotating the hand rotates the object. “Scale” also works by holding the pinch, when the user moves upward the object becomes bigger and when the user moves downward the object becomes smaller. The remove item does exactly what the label describes, on pinch the object will be removed from the virtual environment. Requirement 7 is met with this menu system and menu items.

3.5 Adding objects

Interaction with existing objects can be realized with the input and menu system described in the previous sections, but adding new objects requires an additional system. In [39] and [30], a panel is used for each user to enable interaction for the user holding the panel. The IPanel is inspired by this concept. The IPanel is a virtual panel mounted on the table in the virtual environment of the Digital Maquette system. Figure 13a shows an empty IPanel. The panel is always present in the environment on a fixed location. The panel is visible for every user and every user can interact with this panel. This should make adding objects to the environment understandable to all users. The users will never have to search for the IPanel because it is at a fixed location, which makes the environment simpler. A simpler environment aid users to understand the interface better and learn to control it faster. Two buttons are displayed on the panel; the previous button and the next button, when pressed, the panel displays the previous or next predefined object. This new object can be moved like any other virtual object. If the object is moved, scaled or rotated it is decoupled from the panel making it part of the environment. Because the panel only has two buttons the users should be able to learn how to control the panel relatively fast.

In Figure 13b an IPanel is displayed with a cube attached to it. The cube will be automatically placed in the environment when the user interacts with it. This is could be an intuitive way of placing objects in the environment because it does not require additional actions or buttons. With the IPanel and the menu system, requirement 6 is met because the IPanel makes adding objects possible and the menu system makes removing existing objects possible. To give the users extra information about the object which is currently attached to the IPanel a short piece of text is displayed. The text on the IPanel corresponds to the name of the object. Because the IPanel is mounted at a fixed location on the table, the text is displayed on the two longest sides of the panel. This way the text is also readable for users that view the environment from the other side of the table, which is an important feature for the usability with concurrent users.



(a) Empty (b) Cube

Figure 13: The IPanel

3.6 Collaboration

In order to work collaboratively with multiple users, users have to understand what other users are doing. Otherwise users might become confused if, for example, an object starts moving without notice. In the real world people can see the actions of other people. When actions of others are visible, a person is able understand what another person is doing. This can be mimicked in the Digital Maquette interface. To interact with the interface the hand of the user is used as input device. The hand is tracked, so the position and orientation are known. This means that the hand of a user can be displayed in the interface of another user. Displaying the hands of users lets users view other users actions. Viewing the actions of others gives users the ability to understand them. Therefore, Requirement 5 and 8 are met with this property.

3.7 Networking

Supporting multiple users in different rooms or buildings, requirement 5, requires networking. Networking itself is not part of the interface, but the networking has certain restrictions which are important for the interface. A small description of the networking system is given in this section to illustrate the properties of the network.

The network communication is done through an IP network, illustrated in Figure 14. The network consists of a number of replica-managers. These replica-managers communicate through each other by broadcasting UDP packets. A client connects with TCP to one replica-manager and retrieves the committed updates from that replica-manager. When a client wants to change an object, it sends the proposed update to the replica-manager. With a voting technique the replica-managers commit or drop the proposed update. When an update is committed on a replica-manager, it is propagated to all clients of that replica-manager.

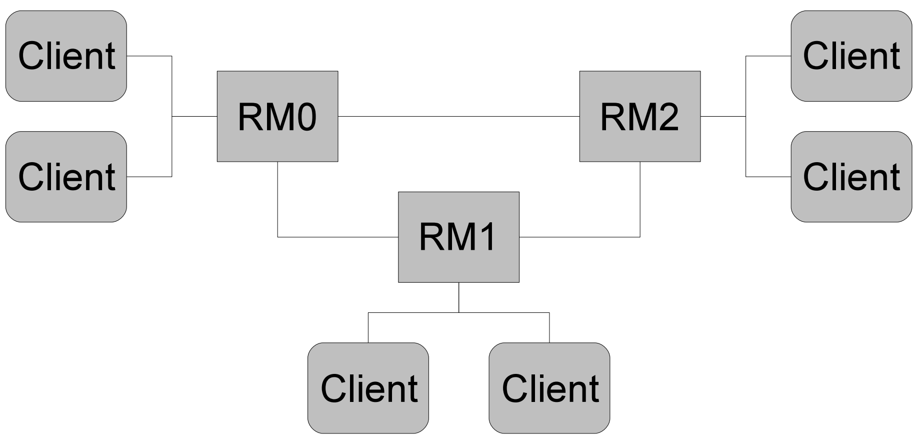


Figure 14: Network structure

The important properties of the network for the interaction are:

- Not all proposed updates from a client are committed;
- A client receives no notice if an update is dropped;
- A client defines updates based on its cache. This means when the cache is outdated the update will never be committed;
- A client renders from a cache which is possibly outdated.

These properties are important for the interaction of the interface. To account for the first two properties, the interaction must be designed in such a way that, when an update is not committed, the user interface must continue to function. This means that the interaction system must never wait for an update, because there is no notice when an update is not committed. The third property defines that an update formed by the client (interface) can be based up on outdated cache and, therefore, never be committed. To maximize the possibility of the up to date cache, the updates should not be issued faster then the network can negotiate and propagate. To create the maximum amount of time between the update, the interface should only send one update per interaction event. An interaction event is triggered when a user moves the hand which is tracked. The hand tracker and hand pose estimation deliver circa 15 input events per second. This leaves the network to negotiate and propagate in approximately 66ms before the interface can issue another update. With this technique, the issued update by the interface has the maximum possibility of being committed which improves the responsiveness and, therefore, the usability. The fourth and last important property is the cache of the client. This property is mentioned because is does affect the interface. This property defines that, for example, an user may look at an object that might actually be moved or removed by another user. The interaction system of the interface must honor these network properties in order to keep the usability high.

3.8 User awareness

Users have to be informed of the ongoing events in the environment. Events can be, for example, that a user joins or leaves, information about networking problems. User interfaces tend to inform their users with messages by a popup, however this method is not usable with this kind of user interface because a popup interferes with the interaction. Games solve this with a HUD (head-up display), for example in Quake 4 [20] and Unreal Tournament 3 [12]. Figure 15 is an illustration of what the HUD looks like in these games. The concept of a HUD system can be used in the user interface of the Digital Maquette system because it does not involve any interaction. The HUD in the user interface is basically a text box in the top part of the screen. A new message will be placed at the bottom line of the text box making the existing text scroll up. This will always give the user the latest messages on screen, keeping the user informed of all events. In Figure 13 the HUD is displayed as a blue semi-transparent box.

(a) Quake 4, from [20]   (b) Unreal Tournament 3, from [12]

Figure 15: HUD systems in games

3.9 Summary

In this chapter the collaboratively 3D interface is described; what it must do and what it should not do. The most import property is the usability of the interface because the interface is aimed at common people. In order to achieve a high usability the interface must be easy to understand. In the design of the interface simplicity is central and, therefore, the interface should be easy to understand without any foreknowledge.

The user can use one hand to interact with the Digital Maquette system. With the input of the hand, two basic forms of interaction is possible. Having only two basic forms of input makes interacting simple. With these forms of interaction, the menu system can be controlled.

With the menu system it is possible to change the location, orientation and scale of an existing object. Removing an object is also done through the menu system. The menu system is part of the virtual environment and every user can see the menus. With the ability of seeing the actions of others users, users have the ability to understand what is happening in the environment.

Adding an object is made available by the IPanel in the environment. Users can browse through available objects and place an object in the environment by interacting (move, rotate, scale) with it.

The hands of the users are rendered in every connected interface. This makes it possible to pinpoint objects to other users just by pointing. This way the users can work together and understand the actions of each other.

4 Realization

This chapter describes the implementation of the concept and the problems encountered during implementation. Before implementing a system, a proper programming language is required. In this situation speed is required to process, for example, the webcam feed. Because the Digital Maquette system is a prototype and research system, the program should be easy to adjust. Performance can be achieved by using C. However, with C, time is required to deal with garbage collection and pointers. Another disadvantage of C is that is does not have a large standard library. Not having this requires extra implementing time in comparison to a language which does have a large standard library. This is why a combination is chosen, Python and C. Components that have to be fast can be programmed in C and the rest of the system can be created in Python. Python does have a large standard library which makes adapting easy.

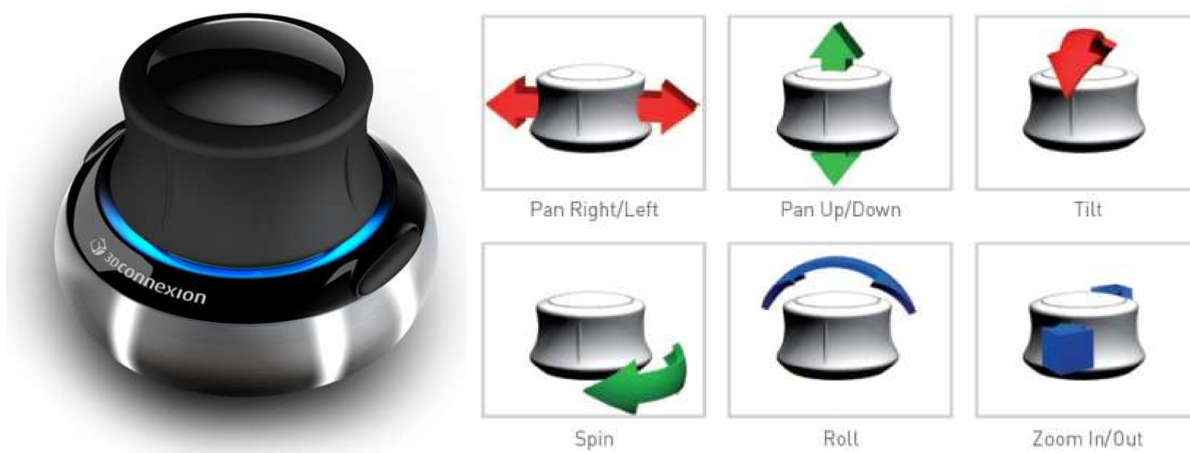
The development platforms are Linux and OSX due to easy development on these platforms. Because Python supports both platforms only the C components have to be ported. Porting the C components however, should not take to much effort because Linux and OSX are both Unix-based systems. In most cases, the C program code does not have to change, only the compile and linking flags have to be adapted.

The virtual environment has to be displayed in the HMD of a user. To display the 3D environment, OpenGL is used because it is an open standard and is platform-independent. Python and C both have libraries to use OpenGL and, therefore, OpenGL seems a suitable solution.

In the next paragraphs, problems that arised during the implementation of the prototype are described.

4.1 Input

In order to let the user interact with the system, the hand of the user is used. To be able to know the hand pose, two separate components are needed, hand tracking and hand pose estimation. The hand tracker searches for the hand in two video feeds (not from the HMD mounted webcam) and sends the result to the hand pose estimation. The hand pose estimation then maps the virtual hand on the result of the hand tracker. This virtual hand is used for input to the system and is also displayed to the users. The hand tracker is created by M. Fremouw and the hand pose estimation is created by G. Boer. For more details on these systems see [15] and [4], respectively. While developing the system these components where not available yet, this means that the input has to be simulated in order to be able to test any interaction. The first approach of simulated input is realized by using the keyboard. Three keys are mapped for X, Y and Z movement. This approach is far from ideal, even very experienced users (programmers) were constantly confused which key they needed to press in order to activate the desired movement. The second approach realized by using the mouse to move the virtual hand, because the mouse only has 2 DOF the mouse is only able to move at two axis. This is fixed by changing one axis when the right mouse button is pressed. The mouse can be used to move along the X and Z axis, enabling the user to move left, right, away and closer. When the right mouse button is pressed, the Z axis is swapped with the Y axis. This enables the user to move the hand up and down. This does not cope with rotations at all, but this kind of input is enough to be able to develop the biggest part of the system. To simulate real 3D input, a different input device is required. A 6 DOF input device is required to be able to cope with the possible movements and rotations. The SpaceNavigator [2] from 3DConnextion is such a device. Figure 16a shows an image of a SpaceNavigator. This input device can pan, tilt, spin, roll and zoom at the same time, in Figure 16b an illustration of the input methods are shown. This device gives the ability to fully develop the user interface and interaction system without input from the users' hand. To move and rotate the hand in the environment, the pan right/left, pan up/down and zoom in/out is connected to the X, Y and Z axes of the hand, respectively. In order to rotate the hand, tilt, spin and roll are connected to the rotation along the X, Y and Z axes, respectively.



(a) SpaceNavigator (b) Input methods

Figure 16: 3D Connexion - SpaceNavigator, from [2]

4.2 Networking

The concurrency of the system depends on the networking ability of the system. This component of the system is created by H. Lenting [25]. Integrating the networking system is not trivial at all. A user interface generally changes data of an object and builds upon that change. With a distributed system this is not possible, the user interface has to propose updates. The user interface cannot even wait for an update to be committed because the update can be dropped just as easily. It is a difficult task to decouple the user interface with the storage in the way that is described in the concept. While developing, a dummy networking layer was used. With the dummy layer, updates are always committed. Another property of the dummy layer is that there is no delay in the commit, because the update data is written immediately in the storage. In one interaction event it is possible to use the data of a previous commit when there is no delay in a commit. Using this aspect of the dummy layer should be avoided because with the normal networking layer this is not the case.

When the user interface is attached to the network for the first time, a lot of updates where dropped by the network. The updates were dropped even on slow/normal usage. This resulted in objects occasionally not being displayed. This effect was caused by the user interface sending more than one update in one interaction event. These type of situations are typically hard to fix, because they occur occasionally. The updates are now reduced to one per object per interaction event, this makes the interaction system very robust in terms of distributed networking.

With the networking system fully integrated with the user interface, the ability to move the hand on multiple displays is established. When a user tries to move his/her hand, the interface proposes an update to the distributed network. This proposed update is approved by the network or becomes obsolete over time. On approval, the update is sent back to the user interface. When the update cannot be approved, the proposal disappears when a committed update makes the proposal obsolete. In other words, when the user sees the virtual hand moving on his/her display, the network has already approved the movement and pushed the "move" action to the interface of the user. This approach immediately raised the question: Is the latency of the network in the user interface low enough? When the latency is too high, a solution would be to move the hand smoothly in the users' own interface and not wait for updates from the network. This has the drawback that the data of the hand object is not consistent with the distributed network. With this drawback comes another problem, the updates proposed on outdated data are dropped by the distributed network. So decoupling the hand object to move smoothly is not a solid approach. But in preliminary tests the latency of the network was actually low enough. Even with an artificial 30% packet loss the hand moves reasonable smoothly. The responsiveness of the interface will have to be experimented on to find out where the border between acceptable and problematic lies.

4.3 Models

A crucial ability of the system is to load models from the disk, not only because it is very complex to create a model from code but, more importantly, architects need to load their existing models. Wavefront developed OBJ [37], an open format for exchanging objects. This format is supported by many 3D applications, which makes it a suitable choice to embed in the Digital Maquette system. Models can have many vertices, when the number of vertices increases the time required to render the model also increases. Relative complex models can affect render speed in such a way that it is not workable anymore. For example, a model with 84784 vertices takes approximately 35 seconds to render. To create an acceptable situation, the model rendering has to be optimized. A faster way is to create an OpenGL display list for every object. This display list is only for the geometry of a model and is stored on the video card, in other words, all the vertices and normals are stored in a display list on the video card. This makes it possible to render the geometry of an entire model with only one call to the video card, which has an enormous performance boost. Because the environment can have multiple objects of the same model, these objects can share the same display list. This property makes it possible to create display lists of the models at program startup, making it unnecessary to load data from disk when an object is placed in the environment. Preloading the data results in no loading delays when users are interacting with the system.

4.4 Summary

This chapter describes the problems and solutions encountered while implementing the Digital Maquette system. While implementing the interface, no hand input nor networking was available. In order to develop the interface some sort of input is required to simulate the hand. The hand is simulated by three different input devices, keyboard, mouse and the SpaceNavigator. With a simulated hand input, it is possible to interact with the interface while the hand input is not yet operational.

During the development of the interface the networking layer was replaced by a dummy layer. This gave the opportunity to create the interface apart from the networking. When the networking was ready, it had to be integrated with the interface. The integration of the networking layer is not trivial and has to overcome some hard problems.

When the Digital Maquette system is realized, preliminary tests are conducted by the developers. These tests are executed to ensure that the implemented functionality works correctly. During these tests, many small problems where addressed. In order to evaluate the usability of the Digital Maquette system, experiments with test subjects (users) that are not involved in the developing stage are required. In the next chapter experiments are defined to evaluate the realized prototype.

5 Experiments

In order to evaluate the Digital Maquette system, several experiments have been performed. This chapter describes the experiments and will summarize the results. The experiments are important because they check if the quality of the Digital Maquette system is adequate for end users. The evaluation is divided in three experiments, which evaluate the intuitiveness, the understandability of the concurrency and the responsiveness of the Digital Maquette system.

To understand in what conditions the experiments are run, the conditions and circumstances of the experiments are defined in the next paragraph. After the conditions the experiments are defined.

5.1 Experiment conditions

It is important to define clear conditions in order to keep the experiments unbiased. The experiments are designed to evaluate the interaction with the interface of the Digital Maquette system, focussed on intuitiveness, understandability of the concurrency and the responsiveness. It is important to eliminate other aspects of the Digital Maquette system because that can lead to a biased user. Therefore, in the experiments, users do not wear video glasses but see the output of the Digital Maquette system on a screen. Video glasses are not common in consumer electronics, using these with the experiments may lead to a user that is only focussed on the glasses and not on the tasks defined in the experiments. Another reason to not use the glasses is that they require a video feed for background video and to detect the marker. This would introduce an extra bias because users would have to look at the table in order to see the interface of the Digital Maquette system. With the output on a screen, the user has a static view of the Digital Maquette system, which again makes the user more focussed on the task. To eliminate uncontrolled behavior of the networking part, the networking part is reduced to one replica-manager, which is basically a client-server network. One replica-manager does not negotiate over UDP and bypasses the replication part of the networking, making the Digital Maquette system simpler and the results more focussed on the interface. The input of a user is changed from the users' hand to a mouse, this has two reasons: The hand tracking and pose estimation are not ready at the time of the experiments and the user can be biased when using his hand for input because the virtual hand would move with the real hand, which may be distracting.

To gather data from the users, every interaction with the Digital Maquette system is logged. This is done by a special client (the recorder client), connected to the replica-manager. The recorder client records to a file, which is stored separately for every session. A recorded session can be viewed by another special client, the play client. With the play client it is possible to display what users see while interacting with the system. In the experiments, vocal communication is possible. Therefore, all vocal communication during the sessions is recorded. With both, session and vocal recordings, enough data is available to replicate the experiments. The ability to replicate the experiments is important because otherwise, if extra data is required in order to come to a conclusion after an experiment, that experiment would have to be executed again.

In the experiments, a maximum of two concurrent users is set. Therefore two computer systems are required. The following computers are used for the experiments:

- Dell XPS M1530
  The operating system is Ubuntu 8.04 Hardy Heron running on an Intel Core Duo CPU T7250 at 2.00GHz with 2GB memory, a Marvell Technology Group Ltd. 88E8040 PCI-E Fast Ethernet Controller and an nVidia Corporation GeForce 8600M GT video card with a 15" TFT screen.
- MacBook Pro 15"
  The operating system is Ubuntu 8.04 Hardy Heron running on an Intel Core Duo CPU 2.16 GHz with 2GB memory, a Marvell Yukon Gigabit Adapter 88E8053 Singleport Copper SA Ethernet controller and an ATI Radeon X1600 Mobile video card with a 15" TFT screen.

These two computer systems are reasonably similar in terms of performance and can, therefore, be used in the experiments. When networking is required in an experiment, the computer systems are connected through a 100mbit direct link. With a direct link there is no additional Ethernet traffic that can affect the experiments.

Before a user starts with the experiments, a short explanation of the goals of the Digital Maquette system is given to give the user an idea of its purpose. This explanation is not told by the supervisor but given in the form of a printed text to avoid bias. Along with the explanation, a description of the mouse input is also given because using the mouse as a 3D input device is relatively complex and not part of the interface. After the user reads the description, one of the supervisors also explains the mouse input with the use of gestures, to make sure the user understands the input method.

In order to have an impression of the experience of the user, a number of questions are asked before the experiments start. The following questions are asked:

1. What is your age?
2. How many times do you use a computer?
   - O About 1 to 3 times a year
   - O About 1 to 3 times a month
   - O About 1 to 3 times a week
   - O Every day
   - O Two times a day or more frequently
3. Are you an experienced computer user?
   - O No, I am not an experienced computer user
   - O I know just enough to do my work
   - O The basics of computer usage is clear to me
   - O My knowledge of computer usage is above average
   - O I am a computer expert
4. Do you have experience with three-dimensional images or worked with them before?
   - O No, never seen or worked with them
   - O Did see them, but never worked with them
   - O Have worked with three-dimensional images
   - O My experience with three-dimensional images is above average
   - O I am an expert in this area

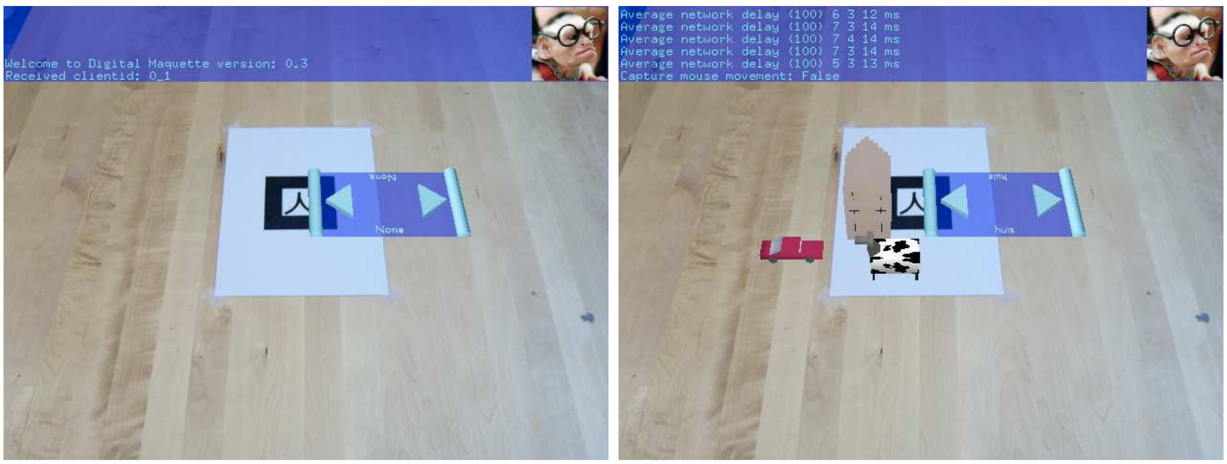
With these questions the experiences of the users is made clear, also the age is asked to see if there is a relation between the age and the learning curve.

In the next three paragraphs the experiments are described.

5.2 Experiment 1: Intuitiveness

This experiment is used to find out if users can easily discover the functionality of the 3D collaborate interface. In this experiment users are asked to perform a simple task, the time spent on the task is measured. With the questions asked and the time spent on the task an impression of the intuitiveness of the interface should be available.

The user is asked to create a landscape with a car, a house and a cow. Figure 17b shows the environment with this task completed. With this task, the user is required to use all interaction aspects of the Digital Maquette system, create objects, use the menu and move objects. If the user manages to get this task done, the interface should be reasonable clear to the user.



(a) Initial (b) Completed

Figure 17: Setup experiment 1: Intuitiveness

Initially, only the IPanel is shown, as displayed in Figure 17a, that allows users to create objects. The users work in their own disconnected environment. Two users are asked to finish the task at the same time and, therefore, can ask each other questions. The networking ability of the Digital Maquette system is not used because this experiment focusses on the intuitiveness of the interface.

When the user is done, either 30 minutes have passed or the task is finished. The user is then asked to answer the following questions:

1. Is it clear how to use the program?
   - O Completely unclear
   - O Not clear
   - O Reasonable clear
   - O Almost clear
   - O Completely clear
2. How did you find it to discover the functionality of the program?
   - O Very difficult
   - O Difficult
   - O Neutral
   - O Easy
   - O Very Easy
3. What did you think of the reaction speed of the program?
   - O Very slow
   - O Slow
   - O Neutral
   - O Fast
   - O No delay noticed

These questions give the user's perspective of the experiment and can be used to conclude if the Digital Maquette system is intuitive. An additional question is asked to check how much the user understands of the interface of the Digital Maquette system:

4. Can you explain how the program works?

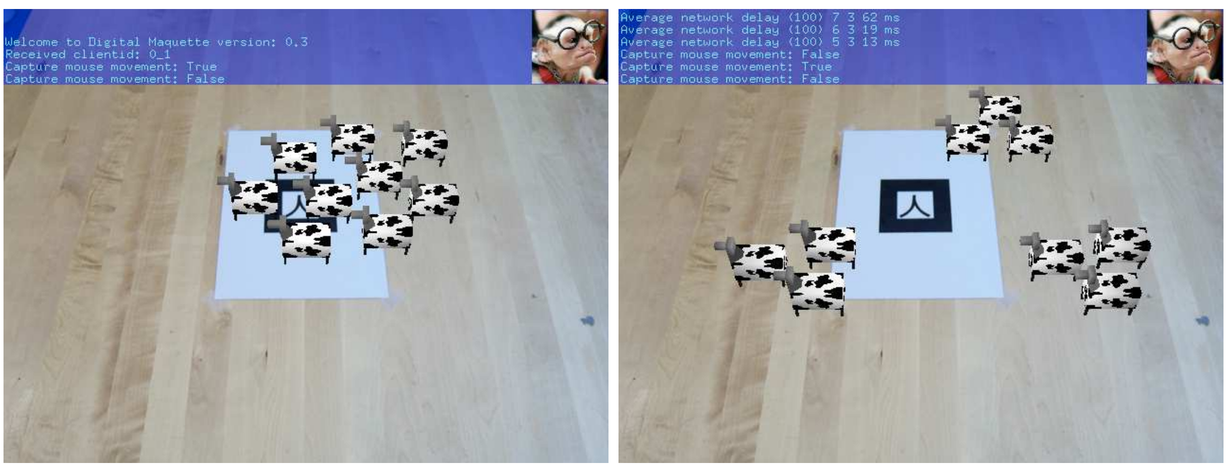
This should give insight if the user actually knows how the Digital Maquette system works, because the users might think they know the Digital Maquette system while in fact they do not.

5.3 Experiment 2: Concurrency

The Digital Maquette system supports multiple users, actions executed by other users can be difficult to understand. This experiment is used to find out if the users understand the concurrency of the Digital Maqette system. In this experiment the time the users take to complete the task is measured.

The users work in pairs and the task is concurrent. Nine cows are placed in the virtual world and the users are asked to re-arrange these cows in three groups of three cows.

Both users have the same task and work in the same virtual environment. This means the mouse movement of both users is displayed on both screens. The IPanel of the first experiment is removed to keep the users focussed at the task. For all users, the nine cows are placed at exactly the same positions, to keep the environment equal for all users.



(a) Initial (b) Completed

Figure 18: Setup experiment 2: Concurrency

The experiment ends when the users complete their task or 30 minutes have passed. After the experiment the users are asked to answer the following questions:

1. Was it clear what the other user was doing?
   - O Completely unclear
   - O Not clear
   - O Reasonable clear
   - O Almost clear
   - O Completely clear
2. Was viewing the actions of the other user distracting?
   - O Always distracting
   - O Sometimes distracting
   - O In some situations distracting in other not
   - O Mostly not distracting
   - O Not distracting at all
3. How did you find completing your task while another user was active?
   - O Very difficult
   - O Difficult
   - O Neural
   - O Easy
   - O Very Easy
4. Does the program respond fast enough to work concurrently?
   - O To slow, not workable
   - O Very slow, badly workable
   - O Reasonable fast
   - O Fast, very workable
   - O No delay noticed

These questions should give an insight of the experience of the users during the experiment. To add extra detail several open questions are also asked:

5. During the experiment, did something distract you and if the answer is yes what did distract you?
6. Did you find it difficult to create the groups? Why?
7. Do you think that in certain situations it could be handy to work with multiple concurrent users?

   Yes? Which situations, and why would it be handy?

   No? Why can't it be handy?
8. When working with multiple users concurrently, is it crucial to be able to talk to each other?

The additional questions are important in order to retrieve the opinion of the users about the concurrency.

5.4 Experiment 3: Responsiveness

An interactive interface needs to have a certain amount of responsiveness to let the users work in an optimal way. This experiment is to find out how fast the Digital Maquette system should react in order to keep a workable interface. The experiment starts with no extra network delay and with every step the delay is increased with 50ms up till 500ms. At every step the user is asked to give feedback about the responsiveness of the Digital Maqette system.

The users are asked to do a continues task: move a car continuously from the left side to the right and back again. This task is executed in a disconnected environment, both of the users have their own environment. All other objects are removed except for the car.

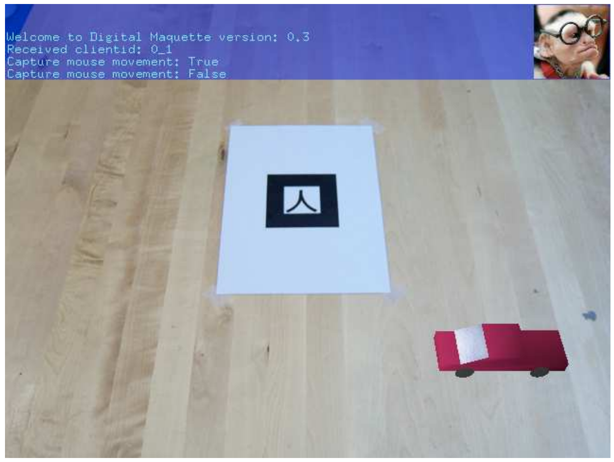


Figure 19: Setup experiment 3: Responsiveness

In every step the user is asked the following question:

1. What did you think of the responsiveness of the system?
   The reaction time of the system was:

   O Not workable

   O Quite slow

   O Workable

   O Good

   O Very good

When the user reports “Not workable” the experiment ends, it has no use to do additional steps because the answer would continue to be “Not workable”. In any case, the experiment does not go any further when it has reached a total delay of 500ms. In preliminary tests 500ms is defined as extremely slow, which makes 500ms suitable for the upper bound of this experiment.

In the next chapter the results of the three experiments are described.

6 Results

The previous chapter defined three experiments, intuitiveness, concurrency and responsiveness. In this chapter the results of these experiments will be described. In the first paragraph the experience of the users in the field of 3D applications will be described, this is important to get an insight into what kind of background the users have that follow the experiments.

6.1 User Information

There are many different users, with and without experience in 3D applications. To be able to test the intuitiveness of the interface people are picked that do not have much computer experience. People that have experience in 3D applications might have expectations of the interface and for this reason might be biased. For the concurrency experiment, people with experience of concurrent networked systems might also be biased because they can also have expectations. Therefore, inexperienced people are used to perform the experiments.

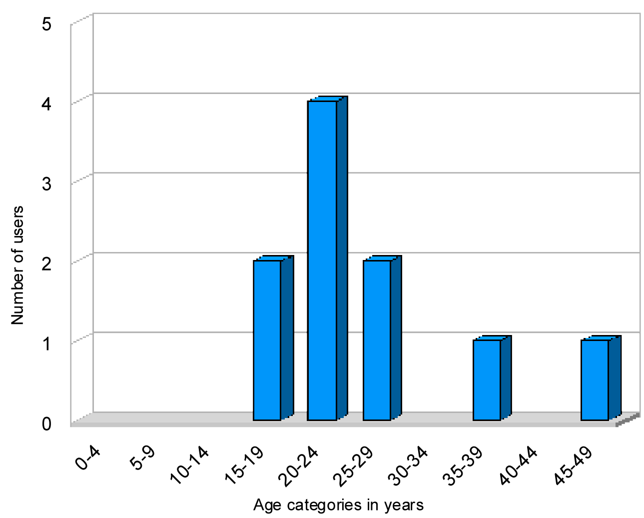


Figure 20: UIQ1: Age of users divided into categories

Finding inexperienced computer users can be problematic, but Kinderdagverblijf Us Twadde Thús lent ten employees for the experiments. Kinderdagverblijf Us Twadde Thús is a company which operates in the childcare sector. The employees care for children from 0 years old to 12 years old. The employees in general have no advanced computer knowledge based on their jobs but to check this, questions are asked before the experiments start. All employees are female and most employees are between 18 and 30 years old. For an overview of the age distribution look at Figure 20. Kinderdagverblijf Us Twadde Thús has a registration system that all employees use. Therefore, the question "How many times do you use a computer?" is answered with "Every day" or even more frequent (Figure 21). This means that the users have experience using a mouse, which is used as input device of the Digital Maquette system during the experiments. The next question asked is "Are you an experienced computer user?" Most users answered with "The basics of computer usage is clear to me" and a minor part answered with "I know just enough to do my job" (Figure 22). The users performing the experiments are used to use a computer but do not have advanced computer experience. To see if the users have experience with 3D applications the question "Do you have experience with three-dimensional images or worked with them before?" is asked. Most users answered that they did see 3D images before but never worked with them or did not see them before at all. Only 30% of the users have worked with them (Figure 23). This means that the experience of the users with of 3D environments is not very high.

The experiments can start now that the background of the users is known. In the next paragraphs the results of the experiments are described.

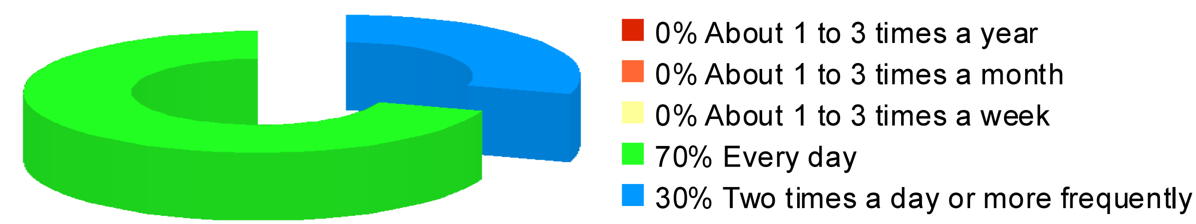


Figure 21: UIQ2: How many times do you use a computer?

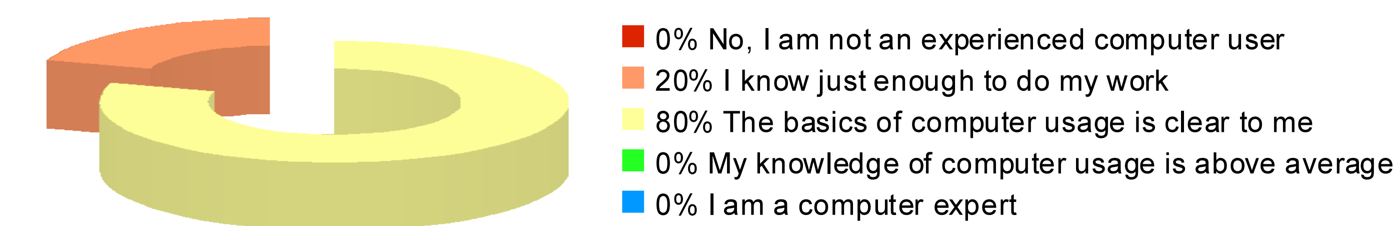


Figure 22: UIQ3: Are you an experienced computer user?

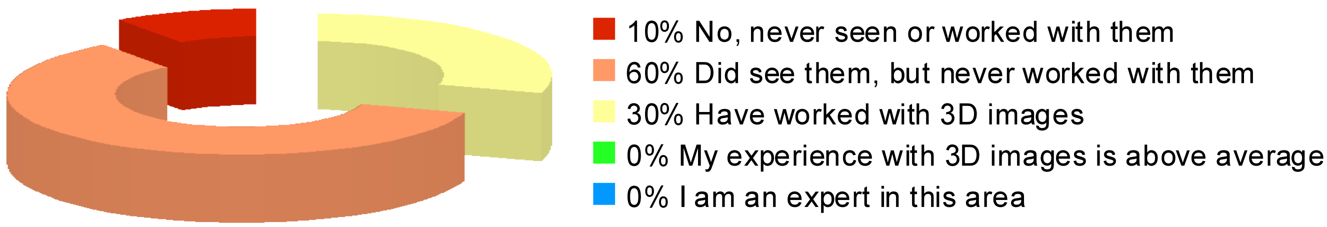


Figure 23: UIQ4: Do you have experience with 3D images or worked with them before?

6.2 Experiment 1: Intuitiveness

To measure the intuitiveness, the time is used that the user needs to complete the task. Figure 24 is a graph of the time required by the users to complete the task. Every user finished the task. There was one user who required 15.3 minutes to complete the task and two users required roughly 11 minutes. The rest of the users completed their task in under 6.5 minutes. All the users tried to click on the buttons of the IPanel, much the same like a button on a webpage. In general, the users had problems with the depth of the interface. In the beginning of the experiment no user moved the mouse up or down only sideways and to the front and back. After moving the mouse around a bit, users read the description of the mouse as input device again. Reading it again made the users use the depth of the system and figuring out how the buttons of the IPanel work. When the users understand the buttons of the IPanel, creating the landscape took less time than understanding the input.

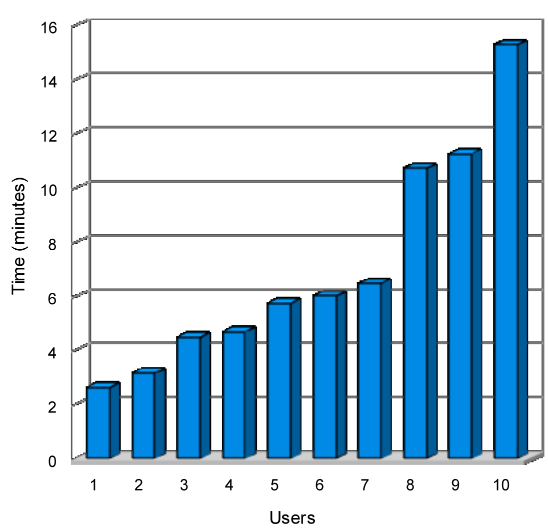


Figure 24: E1: Time required by the user

After completing the task the questions were asked. The first question is “Is it clear how to use the program?", the answers are displayed in Figure 25. Various answers were given to this question, one user did not understand the interface at all and one user did not find the interface clear. Most users find the interface reasonable clear or even more clear. To check the clarity of the interface the following open question is asked: "Can you explain how the program works?" The answer of this question is compared with the first question, for most users the answers correlated. But two users answered the first question with "Reasonable" and could explain the usage of the interface completely. Therefore, the answers of the first question might be more positive than the graph in Figure 25 shows. The next question asked is "How did you find it to discover the functionality of the program?", Figure 26, most users answered "Mediocre" and two users answered "Difficult" Which is peculiar because the two users that answered with "Difficult" were two of the fastest users in the group (second and third). In order to have an impression of the reaction speed of the program the question "What did you think of the reaction speed of the program?" is asked. The results of this question is displayed in Figure 27. One user answered "Slow" and two users answered "Mediocre" but all other users answered either "Fast" or "No delay noticed"

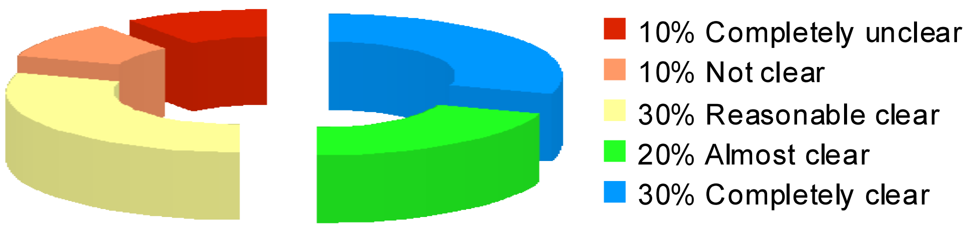


Figure 25: E1Q1: Is it clear how to use the program?

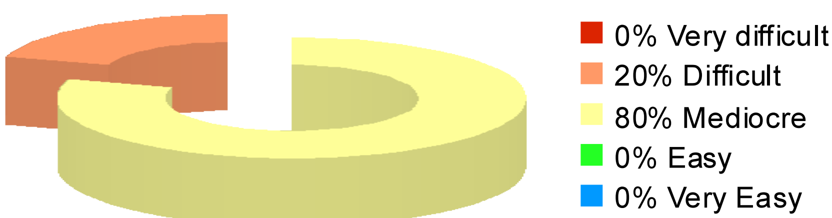


Figure 26: E1Q2: How did you find it to discover the functionality of the program?

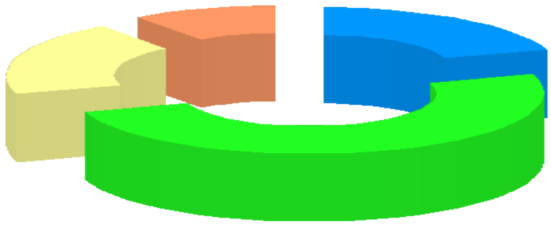


Figure 27: E1Q3: What did you think of the reaction speed of the program?

6.3 Experiment 2: Concurrency

To measure the understandability of the concurrency, the time required to complete the task is measured and, after the experiment, questions are asked. Figure 28 is a graph of the time required by the user pairs. Because the task is completed in pairs, time is measured for a user pair instead of a single user. All pairs finished in under 3.6 minutes and most pairs finished in under 2 minutes. The two users that needed the most time in the concurrency experiment also needed the most time in the intuitiveness experiment. These two users did not understand the task completely. They thought they where ready while they still needed to move cows to complete the task. Another problem during the experiment was the mouse. The mouse can move outside the viewport which happened in three cases. If the mouse is "lost" the client needs to reset to display the mouse in the center of the viewport. In this experiment, one user had a problem with the depth of the Digital Maquette system. This user did not realize that the mouse was floating above the cows in the environment. During the experiment, users communicated with each other, four pairs made agreements about which cows to move. While the users were busy with relocating the cows, in general no user wanted to grab a cow where the other user was working. In only one case the user pair had a small "fight" about a cow in the environment (the last cow). In two cases, the majority of the cows were relocated by one user while the other user was only moving the mouse and not any cows. In these two cases the user moving the cows encouraged the other user to move a cow. One of the encouraging users even explained how to move a cow.

After completing the task, the defined questions were asked. The first question asked is "Was it clear what the other user was doing?", the answers are displayed in Figure 29. This question is answered by every user "Reasonable clear" or more clear. The second question is "Was viewing the actions of the other user distracting?", in Figure 30 the results are displayed. The answers are spit into two groups, not distracting (50%) and distracting (50%). The third question is "How did you find completing your task while another user was active?", Figure 31 is a graph of the answers. 60% of the user found it easy to complete the task in a concurrent environment. The rest of the users answered between "Mediocre" to "Very difficult" The fourth question is "Does the program respond fast enough to work

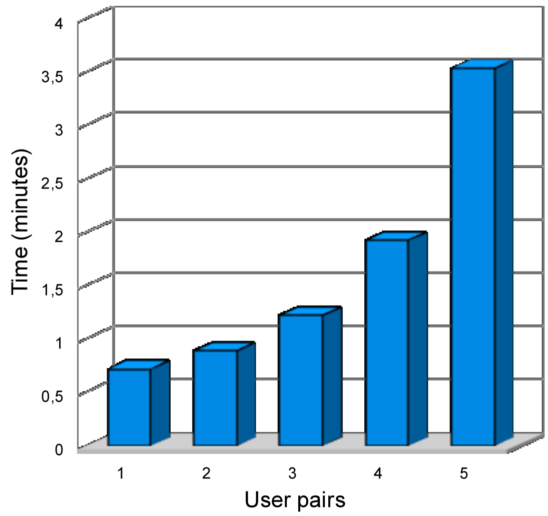


Figure 28: E2: Time required by each user pair

concurrently?", in Figure 32 the answers are displayed in a graph. All users answered that the Digital Maquette system responded "Reasonable fast" or faster, some users did not notice a delay between their input and program output.

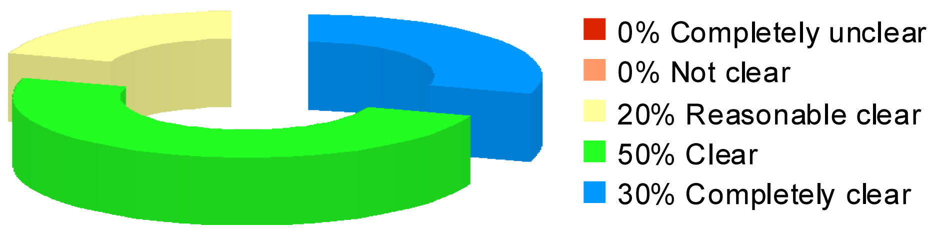


Figure 29: E2Q1: Was it clear what the other user was doing?

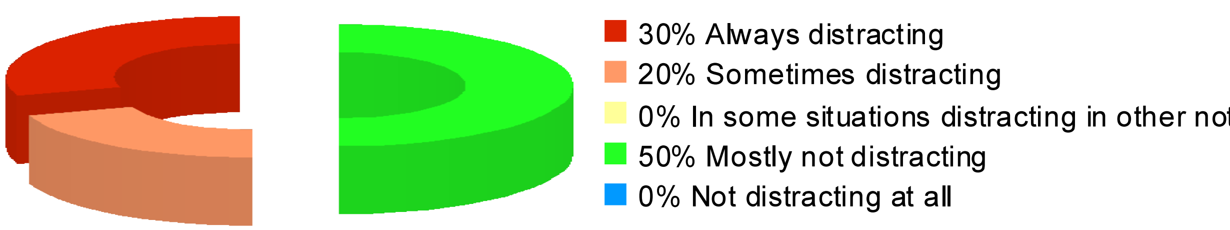


Figure 30: E2Q2: Was viewing the actions of the other user distracting?

After asking four multiple choice questions four open questions are asked. The first open question is: "During the experiment, did something distract you and if the answer is yes what did distract you?" Four users answered that there were no distractions for them.

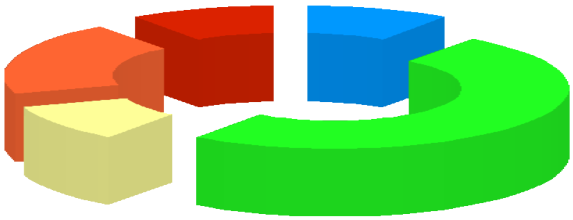


Figure 31: E2Q3: How did you find completing your task while another user was active?

Figure 32: E2Q4: Does the program respond fast enough to work concurrently?

The two users that had a small fight for the last cow answered that it was a bit distracting that the other user was trying to move the cow. One user answered that the mouse cursor was distracting, this was because the user thought that the cursor of the concurrent user was hers. Two users found it a little distracting that the second user moved the cow while the first user wanted to move it. One user found it distracting when the concurrent user opened a menu, this drew the attention of the user. The next question is: "Did you find it difficult to create the groups? Why?" Nine users did not find it difficult to create the three groups because the users were getting used to the interface. One user answered that she found it difficult to grab the cow, this user had problems with the depth of the system. The next question is: "Do you think that in certain situations it could be handy to work with multiple concurrent users?" Two users answered with no, the reason given is: "When you do everything alone, you know exactly what is left to do." and the second reason is: "Working together is distracting." Eight users answered yes, six of them had the same reason: "When you work together you can work faster." The other two reasons were: "When you work on the same design it can be handy." and "Can be handy when you make agreements about who does what." The last question asked is: "When working with multiple users concurrently, is it crucial to be able to talk to each other?" All users answered yes to this question but not all users gave a reason. Four users answered with the reason: "It is easier with the ability to talk to each other." The other three users that gave a reason answered with: "You can tell want you want to do and ask what the other user is going to do."

6.4 Experiment 3: Responsiveness

While the user works with the system, the input of the interface is artificially delayed in steps. Each step the user is asked the following question: "What did you think of the responsiveness of the system?" The answers of this question is displayed in Figure 33. This experiment is conducted on a single replica-manger which means that a delay exists in the networking. This delay is measured during the experiments and varies between 11ms and 17ms in a one client setup. On top of this existing networking delay an extra delay is introduced which is displayed in Figure 33. Without an extra delay the majority of the users answered "Very good" After introducing 100ms extra delay the answers significantly changed, but the users still answered at least "Workable" With an extra delay of 150ms, 20% of the users answered "Quite slow" After 200ms the first users answered "Not workable"

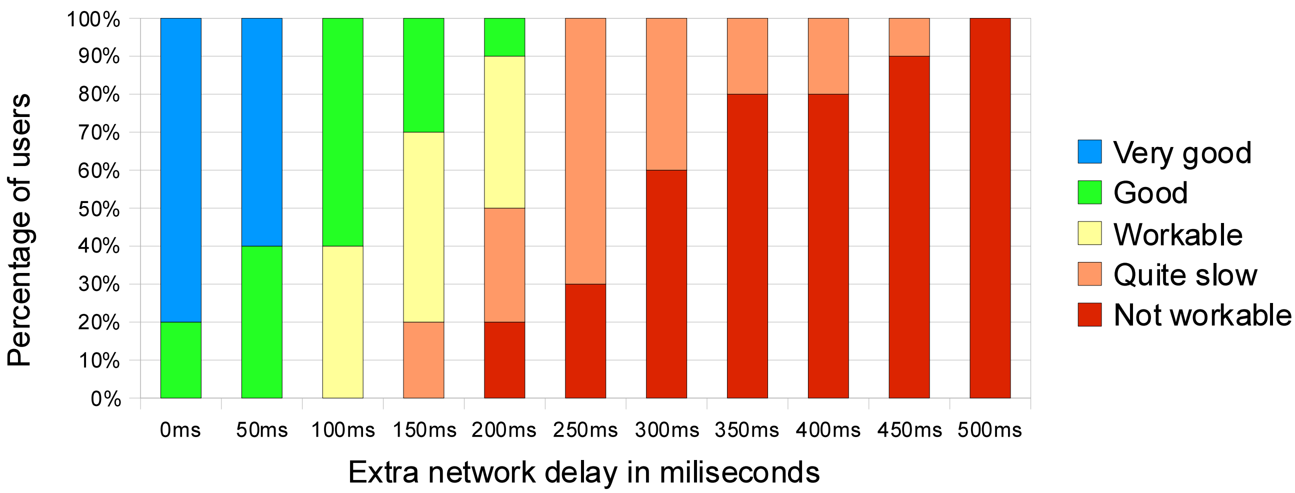


Figure 33: E3: Responsiveness

6.5 Discussion

In this paragraph the results of the experiments are discussed. The graphs of the experiments and the users during the experiments show curious behavior in some occasions. In the next paragraphs these occasions are described and a possible explanation is given.

During the experiments a mouse is used for input (section 5.1), this might not be an ideal input device for the interface of the Digital Maquette system. Even though the input device is described in detail, it is expected that learning to use the input device will take a couple of minutes. During the first experiment, all users tried to click on the buttons on the IPanel. The users are familiar with the classical way of using a mouse and, therefore, clicking on a button might seem a logical approach to the users. When the hands are used for input, pressing the button with a finger might seem more logical, which is the defined interaction to use a button of the IPanel. Only after reading the input description again the users started to use the mouse in all directions.

Experienced users take about one minute to complete the task of the first experiment. Most users did not require more than 6.5 minutes, the average time required by the users is 7 minutes. Removing one minute to actually execute the task, would leave 6 minutes to learn the interface. These 6 minutes includes the time required to learn how the mouse input works. The task itself was relatively simple (placing three objects), therefore, completing the task was expected not to take longer than 15 minutes. These 15 minutes includes time to learn to operate the interface. 6 minutes is less than half of the expected required time. For users that are inexperienced with 3D environments this seems a reasonable amount of time. After completing the task, most of the users responded that they understand the user interface. In the second experiment, the answer that the users understand the interface is evident because the users immediately started moving the cows.

In the first experiment, a question is asked what the user thinks of the responsiveness of the Digital Maquette system. In the third experiment the users are asked the same question again (with no extra delay). A peculiar thing about the answers of these questions is that in the first experiment three users found the responsiveness of the system mediocre or slow. In the third experiment the same three users found the responsiveness very good. The responsiveness of the Digital Maquette system is exactly the same in both scenarios, but the users might experience the responsiveness differently because they understand the interface better in the third experiment. Another explanation could be that it is task-dependent. In the first experiment, the users need to move the cursor in the buttons of the IPanel and move a number of objects. In the third experiment the users only need to move an object. The task is simpler which might effect the responsiveness experience.

Users are asked after the first experiment whether the users understand the interface or not. In general, the users that required more time to complete the task answered that they understand the interface better than the users requiring less time. This might be explained by the following: the users that understand the interface took their time to learn it, the users that were fixed on getting the task done as fast as possible might not really have learned how the interface works.

The expectations of the second experiment, concurrency, is that users could understand each other because they can see each others actions. The results show that the users actually understand the actions of each other quite well. However, a part of the users found it distracting when they see the actions of the concurrent users. This was an unexpected result, but it could be improved. For example, while displaying a menu, the labels could be left out in the interface of users that did not activate that menu. The understandability, however, could suffer from such a measure.

For the third experiment, responsiveness, it was expected that the users would suffer from the latency after an extra delay of 100ms. The users answered that they could work with the interface with a delay of 150ms. This is above expectations. This leaves extra time for the interface to render its content and for the network to transmit the updates. When hand tracking is used, the delay might be even higher. With hand tracking, the user is able to point directly at the end position. With the use of a mouse, the user has to move the cursor to the end position. The user requires to see the cursor move in order to position the cursor correctly. This feedback is not required for the hand and, therefore, the experience might be different.

In the experiments, the users worked with a mouse input system in order to keep them unbiased. There should be an extra experiment that uses the user's hand as input. The hand tracking is not yet operational and, therefore, cannot be used. When the hand tracking is operational, an experiment should be defined that evaluates the interface as a whole.

7 Conclusion and Future work

The goal of this thesis was to design a user interface for the Digital Maquette case. The interface should give the users almost the same experience as a user would have with a physical maquette. This case uses AR to place virtual objects, like buildings, in the real world. When using AR or VR, expensive hardware is often required. This thesis focusses on the use of inexpensive hardware. With the use of inexpensive hardware this system is available for common use. When more people are able to use systems like the Digital Maquette system, people can see the advantages of using such systems.

Using inexpensive hardware limits the design in many aspects. Therefore, designing such an interface in 3D is a challenging task. It is also different from designing a 2D interface in many ways making it even more challenging. Creating interaction on a 3D interface is not straight forward. There are, for example, not many appropriate input devices. The input devices that do exist are not common; and learning how to handle these is often not simple. Using the hand as input device would be a solution. But hand tracking and hand pose estimation are not yet off the shelf. This thesis is part of a larger project together with three other theses. Two of these theses have as goal to make the hand input possible. The other thesis researches networking required to realize collaboration. Networking is required because users must be able to collaborate. Collaboration is an important aspect in the Digital Maquette case because people using a physical maquette also have the ability of collaboration. A 3D collaborative interface based on AR is currently not very common. AR can help users to understand the matter in the virtual environment more than when it is displayed on a traditional 2D screen. Understanding an AR interface, however, can be hard for users. This thesis uses real-life collaboration concepts. In real-life, people see each other and, therefore, are able to understand each other actions. The interface shows users what another user is doing, through showing the hand of that user. This method works well according to the results of the experiments, users understand each other actions while collaborating.

When all the modules of the different theses were designed and built, they had to be integrated with the interface. Integrating all the work was time-consuming, especially integrating the networking part of the Digital Maquette system. The integration was heavily underestimated by all students. In the Digital Maquette project this was something that was learned the hard way. If the modules were integrated earlier in the development stage, the integration would possibly take less time. Because integrating earlier would simplify adapting the modules to work with each other.

There is still a lot of work left to do in this field. The interface defined in this thesis is still in an early stage and not yet ready for deployment because it has to validate with the hand tracker as input. I found it a very challenging task to design such an interface, there were a lot of things that needed to be addressed. Because the four theses are strongly connected, I also learned many things from the other students in the other fields. The process of creating this thesis was very exhausting, but cutting edge techniques are very interesting and, therefore, I really liked creating it.

References

[1] New directions in 3d user interfaces. In *VR '05: Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality*, page 312, Washington, DC, USA, 2005. IEEE Computer Society. Organizer-Doug A. Bowman and Organizer-Bernd Frohlich.

[2] 3D-Connexion. Space navigator. http://www.3dconnexion.com/, 2009.

[3] M. Agrawala, A. Beers, B. Fröhlich, and P. Hanrahan. Responsive Workbench. http://graphics.stanford.edu/projects/RWB/, 1998.

[4] G. Boer. An incremental approach to real-time hand pose estimation using the GPU. Master's thesis, University of Groningen, 2009.

[5] D. A. Bowman and L. F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. pages 35–38, 1997.

[6] D. A. Bowman and C. A. Wingrave. Design and evaluation of menu systems for immersive virtual environments. In *VR '01: Proceedings of the Virtual Reality 2001 Conference (VR'01)*, page 149, Washington, DC, USA, 2001. IEEE Computer Society.

[7] D. A. Bowman, C. A. Wingrave, J. M. Campbell, V. Q. Ly, and C. J. Rhoton. Novel uses of pinch gloves$^{\text{tm}}$ for virtual environment interaction techniques. *Virtual Reality*, 6(3):122–129, 2002.

[8] R. Cavallaro. Integrating virtual and augmented realities in an outdoor application. In *2nd Int'l Workshop on Augmented Reality*, pages 34–47, 1999.

[9] S. Cawood. ARTag. http://geeklit.blogspot.com/2007_08_01_archive.html, 2007.

[10] A. Ceranowicz. Modular semi-automated forces. In *WSC '94: Proceedings of the 26th conference on Winter simulation*, pages 755–761, San Diego, CA, USA, 1994. Society for Computer Simulation International.

[11] P. Dourish and V. Bellotti. Awareness and coordination in shared workspaces. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'92)*, pages 107–114, Toronto, Ontario, 1992. ACM Press.

[12] Epic Games. Unreal tournament 3. http://www.unrealtournament3.com/, 2008.

[13] S. Feiner, B. MacIntyre, M. Haupt, and E. Solomon. Windows on the world: 2d windows for 3d augmented reality. In *UIST '93: Proceedings of the 6th annual ACM symposium on User interface software and technology*, pages 145–155, New York, NY, USA, 1993. ACM.

[14] M. Fiala. ARTag, a Fiducial Marker System Using Digital Techniques. *cvpr*, 2:590–596, 2005.

[15] M. Fremouw. An approach towards real-time hand tracking using standard computer hardware. Master's thesis, University of Groningen, 2009.

[16] M. Green. Color in trademark and tradedress disputes. http://www.visualexpert.com/Resources/color.html, 2004.

[17] A. D. Gregory, S. A. Ehmann, and M. C. Lin. intouch: Interactive multiresolution modeling and 3d painting with a haptic interface. In *Proc. of IEEE VR Conference*, pages 45–52, 2000.

[18] C. Hand. A survey of 3d interaction techniques. In *Computer Graphics Forum*, pages 269–281, Department of Computer Science, De Montfort University, The Gateway, Leicester LE1 9BH, UK; cph@dmu.ac.uk, 1997. The Eurographics Association 1997.

[19] T. Haritos and N. D. Macchiarella. *A mobile application of augmented reality for aerospace maintenance training*. Embry-Riddle Aeronautical University Florida, 2005.

[20] Id Software. Quake 4. http://www.quake4game.com/, 2006.

[21] S. Julier, M. Lanzagorta, Y. Baillot, and L. Rosenblum. Information filtering for mobile augmented reality. In *Proc. ISAR 2000*, pages 3–11, 2000.

[22] T. D. Kler. Integration of the artoolkitplus optical tracker into the personal space station. Master's thesis, University of Amsterdam, 2007.

[23] W. Krüger, C.-A. Bohn, B. Fröhlich, H. Schüth, W. Strauss, and G. Wesche. The responsive workbench: A virtual work environment. *Computer*, 28(7):42–48, 1995.

[24] P. Lamb. ARToolKit. http://www.hitl.washington.edu/artoolkit/.

[25] H. Lenting. Replicating augmented reality objects for multi-user interactions. Master's thesis, University of Groningen, 2009.

[26] L. Optics. Personal displays. http://www.lumus-optical.com/, 2009.

[27] Philips. SPC 1330NC. http://www.azerty.nl/, 2009.

[28] W. Piekarski, B. Gunther, and B. Thomas. Integrating bla virtual and augmented realities in an outdoor application. In *2nd Int'l Workshop on Augmented Reality*, pages 45–54, 1999.

[29] W. Piekarski and B. H. Thomas. ARQuake - Modifications and Hardware for Outdoor Augmented Reality Gaming. *Proc. 4th Int'l Symp. Wearable Computers*, 2000.

[30] D. Schmalstieg, A. Fuhrmann, and G. Hesina. Bridging multiple user interface dimensions with augmented reality. In *Proc. ISAR 2000*, pages 20–29, 2000.

[31] Z. Szalavri. PIP. http://www.cg.tuwien.ac.at/research/vr/pip/, 1999.

[32] B. H. Thomas and W. Piekarski. Glove based user interaction techniques for augmented reality in an outdoor environment. *Virtual Reality: Research, Development, and Applications*, 6:2002, 1993.

[33] L. Vlaming, J. Smit, and T. Isenberg. Presenting using two-handed interaction in open space. In *Proceedings of the 3rd Annual IEEE Workshop on Horizontal Interactive Human-Computer Systems*, Tabletop 2008, October 1-3, 2008, Amsterdam, The Netherlands, 2008.

[34] Vuzix. iWear VR920. http://store.vuzix.co.uk/euro/acatalog/iWear_DV920.html, 2009.

[35] D. Wagner and D. Schmalstieg. ARToolKit on the PocketPC Platform. In *Augmented Reality Toolkit Workshop*, pages 14–15, 2003.

[36] R. Y. Wang and J. Popović. Real-time hand-tracking with a color glove. *ACM Transactions on Graphics*, 28(3), 2009.

[37] Wavefront. OBJ Spec. http://local.wasp.uwa.edu.au/~pbourke/dataformats/obj/.

[38] A. Wilson. XWand. http://research.microsoft.com/en-us/um/people/awilson/wand/, 2004.

[39] M. G. Zsolt Szalavri. The personal interaction panel - a two-handed interface for augmented reality. In *Proceedings of EUROGRAPHICS'97*, pages 335–346, Budapest, Hungary, 1997. Computer Graphics Forum.