Design and evaluation of a new system for VRET for agoraphobia

Master programme Media & Knowledge Engineering
Delft University of Technology

F. S. Hooplort (9362554)
Delft University of Technology
Supervised by Dr. ir. C.A.P.G van der Mast
Delft, September 2005
Abstract:

Virtual reality offers a great new perspective in the treatment of phobias. Virtual Reality Exposure Therapy (VRET) involves exposing a phobia patient to a virtual environment containing the feared stimulus instead of taking the patient into a real environment or having the patient imagine the stimulus. This thesis covers how virtual environments for the treatment of agoraphobia can be implemented based on requirements from therapists. Research has been done on the requirements for the creation of a valid and anxiety-provoking virtual world for the treatment of agoraphobia. Within this thesis a design and evaluation is provided. This thesis also covers the implementation of a “prototype virtual environment”, which can serve as an impulse towards a new framework for VRET of phobias. Finally the used software and techniques will be evaluated.
# Table of Contents

**TABLE OF CONTENTS**.................................5

1 **INTRODUCTION**..........................................................7

1.1 **OVERVIEW**..............................................................7
1.2 **RESEARCH GOAL**........................................................7
1.3 **THESIS OUTLINE**.......................................................8

2 **RESEARCH APPROACH**..................................................9

2.1 **ANALYSIS**..............................................................9
2.2 **DESIGN METHOD**.......................................................9

3 **THEORETICAL BACKGROUND**........................................11

3.1 **VIRTUAL REALITY**.......................................................11

3.1.1 **Overview**..............................................................11
3.1.2 **Immersive and Non-immersive Virtual Reality**..........................11
3.1.3 **VR Technologies**.....................................................12

3.2 **PRESENCE**...............................................................13

3.2.1 **Theories**...............................................................13
3.2.2 **Measurement of Presence**...........................................15
3.2.3 **Causes of Presence**..................................................16

3.3 **ANXIETY DISORDERS**..................................................16

3.3.1 **Panic disorder**.......................................................17
3.3.2 **Agoraphobia**........................................................18

3.4 **VR RET**.................................................................19

3.4.1 **Traditional treatment**.................................................19
3.4.2 **Treatment with VR RET**.............................................20

4 **CURRENT SYSTEM**......................................................21

4.1 **INTRODUCTION**........................................................21
4.2 **TECHNIQUES**...........................................................21
4.3 **CURRENT SYSTEM FRAMEWORK**.....................................23
4.4 **MIGRATING THE VR RET SYSTEM TO WINDOWS XP**....................24
4.5 **CONCLUSION**...........................................................24

5 **PROPOSED SYSTEM**.....................................................25

5.1 **INTRODUCTION**........................................................25
5.2 **REQUIREMENTS ANALYSIS**..........................................25

5.2.1. **Parameters**..........................................................25
5.2.2. **Requirements**..........................................................27

5.3 **TASK ANALYSIS**........................................................28
5.4 **TECHNIQUES**...........................................................29

5.4.1 **Quest 3D**..............................................................30
5.4.2 **Evaluation of Quest 3D**.............................................32

5.5 **DESIGN**.................................................................36

5.5.1 **Modular structure**...................................................36
5.5.2 **Separation of the user interface and the application**...................37

6 **IMPLEMENTATION**.......................................................39

6.1 **VARIABLES**.............................................................39
6.2 **VIRTUAL WORLD**........................................................40

6.2.1 **Method used for converting the Market in Delft to 3D**...................40
6.2.2 **Ambience sounds**....................................................43
1 Introduction

1.1 Overview

Virtual Reality Exposure Therapy (VRET) is an evolving technique that has attracted a lot of research. Traditional exposure therapies (also called exposure in vivo) expose patients, who suffer from a phobia, to real world environments that contain the feared stimulus [7]. E.g. a participant that has a fear of heights is gradually being exposed to higher altitudes on a staircase. Virtual Reality Exposure involves exposing the patient to a virtual environment containing the feared stimulus instead of taking the participant into a real world environment or having the participant imagine the stimulus (exposure in vitro). VRET is an interesting alternative for the more traditional therapies, because it has a lot of advantages in comparison to these traditional therapies. VRET is for example time and cost saving, because the therapist and the participant do not have to travel all the way to the real world anxiety provoking environment. Instead they can simulate this environment, with total control and endless repetition.

It has been proven that VRET is effective for patients with acrophobia (fear of heights), arachnophobia (spider phobia) and fear of flying [45]. The effectiveness of VRET in other anxiety disorders like claustrophobia, fear of public speaking, fear of driving, posttraumatic stress disorder, and agoraphobia also holds promise for the future [7]. At the University of Technology in Delft, in close cooperation with the University of Amsterdam, a generic system for treatment of phobia has been developed, taking into account specific human-computer interaction issues. This system has proven effective for the treatment of certain phobias like acrophobia, claustrophobia and fear of flying [29].

1.2 Research Goal

This thesis focuses on a specific phobia: agoraphobia. Agoraphobia is one of the more complex phobias. It literally means 'fear of open spaces', but later on in this thesis we will see that its definition is much broader than that. Several research projects have been conducted on the efficacy of VRET in treating agoraphobia [3]. One study however showed negative results. The fact that the patients were not able to feel present in the virtual environment and thus didn't experience any feared stimulus, was the main problem in this study [5]. We have done previous research on the parameters that would be necessary to create a valid and anxiety-provoking virtual environment for the treatment of agoraphobia [12]. Through this research we found a set of requirements.

To actually be able to conduct the project itself, the condition that primarily needs to be met is that the needed information is collected, processed and structured. This thesis will suffice to provide the background that is required and will mould the obtained results into a structured report. The problem definition, consisting of several elements, that will serve as a guidance in this thesis is the following:

- Design and evaluate anxiety-provoking virtual environments for the treatment of agoraphobia based on requirements from therapists
- Implement a “prototype virtual environment” for the treatment of agoraphobia, based on these requirements, that can serve as an impulse towards a new framework for VRET of phobias.
- Give an evaluation of the software and techniques that were used to implement the “prototype virtual environment” for the treatment of agoraphobia.

The current VRET system at the Delft University of Technology has a certain framework including hardware and software. All the virtual environments that have been build in the past, were constructed within this framework. The initial goal of this thesis is to research how to implement these parameters into virtual environments for the treatment of agoraphobia, within this framework. Because of the
discontinuation of the software manufacturer during this research, support for the software that was used for this framework has been abrogated. Therefore we decided not to proceed building the virtual environment within the current framework. Instead the future virtual environments should be build within another framework, thus the goal of this research is also to give an impulse towards a new framework. The implemented virtual environment can thus serve as a prototype for the virtual environments that will be built within the new framework.

1.3 Thesis Outline

In this chapter we gave an introduction to the subject of the research and the attended research goal. Chapter 2 will cover our research approach. In chapter 3 we will give some theoretical background regarding the concepts of virtual reality, presence, anxiety disorders and VRET. After this general theory, we will give an overview of the current system in chapter 4. In chapter 5 we will discuss the proposed system. We will give a global analysis, a requirement analysis and task analysis. In this chapter the new techniques, design and implementation of the proposed system will be discussed. After that we will give an evaluation of the system leading to the conclusions and discussion in chapter 6 and future recommendations in chapter 7.
2 Research Approach

There are several research methods on forehead that can be used to approach a research (e.g. Case Studies, Appreciative Inquiry, Questionnaires, Surveys etc.). The first question in determining our research approach is: what should we build? There are four products one can build in the design sciences. Those are constructs, models, methods and implementations. Constructs or concepts form the vocabulary of a domain, a model is a set of propositions or statements expressing the relationships among constructs, a method is a set of steps used to perform a task and an implementation is the realization of an artefact in its environment. In this research, the products that we need to build are those that are necessary to facilitate the design of an anxiety provoking virtual environment for the treatment of agoraphobia. To get to this product we have to go through a few steps. The first step is to analyse the problem domain to get to new requirements. The second step is to determine how to design such a system: a design method. We want to determine how we can evaluate the implemented system in order to draw conclusions [44].

2.1 Analysis

Our analysis consisted of two steps:

- The first step was literature study, collecting all suitable and relevant data on the subject. This meant finding relevant literature, articles, internet sites and other sources of information.
- Secondly we thoroughly analysed the current system framework in order to use its advantages for the new proposed framework and eliminate its disadvantages.

2.2 Design Method

For our design method we used the method as opposed by van der Mast [17]. In figure 2.1 we see the appropriate steps in this design method.

1. The first step is the requirements analysis. To do this we use the information we got from our interaction analysis of the current system and structured interviews. We take the functionality and problems of the current system to determine the requirements of the proposed system.

   We also conducted interviews with therapists to determine their view and desires for the proposed system.

2. Secondly a task analysis is constructed, where we sum up the functionality that the proposed system has to have. This results in a task model.

3. In the third step a global design is drawn up, which meets the specifications.

4. Next the implementation is done, were we also take in to account which technique is going to be used to do the implementation. In this particular case the end product is a prototype.

5. Finally the system will be evaluated. Evaluation is actually done during the whole process. But eventually we have a final evaluation of the prototype by a user. In this case that is the therapist. Our evaluation approach is done through summative evaluation. The basic evaluation criterion of our system is to test whether the new system meets the requirements and specifications and whether the virtual environment can serve as a useful prototype for the future framework. This evaluation will be done by therapists, because they will be the end users together with the patients.
Knowledge and behaviour of users

Workings and organization of current tasks

Task model, requirements and user profile

global design

virtual machine and representation

implementation

prototype

specification

specification

specification

specification

structured interviews

interaction analysis

Figure 2.1 – Human Computer Interaction design process [17]
3 Theoretical background

3.1 Virtual Reality

3.1.1 Overview

There is still some discussion on what Virtual Reality really means. We can tell this by the numerous of definitions that exist. One of the definitions for example is: A human-computer interface in which the computer creates a sensory-immersing environment that interactively responds to and is controlled by the behaviour of the user. [40] Yet another definition is: An artificial environment created with computer hardware and software and presented to the user in such a way that it appears and feels like a real environment. [39]

The term virtual reality is sometimes used more generally to refer to any virtual world represented in a computer, even if it is just a text-based or graphical representation. The term virtual reality (VR) was proposed by Jaron Lanier only a few more than one decade ago. In these years the progression of virtual reality has been tremendous. In the beginning the costs were very high. But as the technology grew, the costs decreased so VR systems were not only affordable for governments. The term VR may be quite recent, the tool however isn’t. As early as 1966, Ivan Sutherland built a HMD (Head Mounted Display) which was connected to the computer. All that it showed was a simple wire-frame cube which could be looked at using the HMD. The first generation of VR platforms that were really immersive, had to be limited to industries and research centres, where the high cost of the hardware and software development was justified. However, at the beginning of the eighties VR was ready to be acknowledged as a feasible technology. In the nineties the technology matured. This resulted in better, more realistic virtual environments and economical systems. Now that the costs of VR systems are decreasing, it is available for more users. There are still things to do for this technology to make it available for everyone. But we are already moving to a situation where VR workstations will be available at work or at home, allowing us to do virtual transactions, shopping, games, trips etc. Because of the interesting aspects of virtual reality and the decreasing costs humans are looking for many ways to apply this technique. One area where virtual reality could be very useful is psychotherapy, which we will discuss later on in this document.

Recently D. Joele, a student from the University of Technology in Delft (Netherlands) did research on the use of Augmented Reality for the treatment of phobias [14]. This research was done in a close cooperation between the Delft University of Technology and MedICLab at the Technical University of Valencia (Spain). Augmented reality (AR) is the use of transparent HMDs to overlay computer generated images onto the physical environment. Precisely calibrated, rapid head tracking is required to sustain the illusion [35]. Augmented reality stands in between Virtual Reality and Reality itself. At MedICLab, Technical University of Valencia, an AR system has been developed for the treatment of phobia of small animals. Up until now only a limited number of patients with fear of spiders and cockroaches were treated successfully [14]. At the moment research is done on the treatment of other phobias as well, using AR.

It is difficult to delimit a general concept of VR that includes all the applications that have been designed up to now under such designation. Nevertheless, it is possible to define VR by the basic concepts that characterize it. Pioneering researcher in this field, Burdea, gave a very useful definition: “Virtual Reality is a complex user interface that includes simulations in real time through multiple sensorial channels. These sensorial modalities are visual, auditory, tactile, olfactory etc.” From this definition we can filter that there are two important characteristics of VR, which we will discuss in the following section.

3.1.2 Immersive and Non-immersive Virtual Reality

First, VR is Immersive, because through special devices the user gets the feeling of being physically present in the virtual world.
Second, VR is Interactive, because the user isn’t visualizing the virtual world in a passive way. The user is actually interacting with the virtual worlds by touching, moving and changing objects.

We can distinguish two types of virtual reality:

**Immersive virtual reality**

Here the user is really immersed into the virtual environment. Most extern stimuli are cut out. This means that the user feels as if he or she is actually in the virtual environment. This is mostly done with an HMD or a CAVE construction.

**Non immersive) Desktop virtual reality**

This is the more conventional way of virtual reality. Here the virtual (3D) world is shown on a conventional (2D) Monitor. The user isn’t immersed into the virtual environment, but can actually navigate and move through the world by using conventional inputs like a (3d) mouse, a joystick or a keyboard.

Another kind of virtual reality that’s worth mentioning is augmented reality. Augmented reality is the use of transparent HMDs to overlay computer generated images onto the physical environment. Precisely calibrated, rapid head tracking is required to sustain the illusion. An observer’s experience of an environment is augmented with computer generated information. Usually this refers to a system in which computer graphics are overlaid onto a live video picture or projected onto a transparent screen as in a head-up display.

### 3.1.3 VR Technologies

There are various techniques used to create Virtual Reality. The HMD and the CAVE are amongst the most popular techniques. In figure 2.1 and figure 2.2 we respectively see images of a HMD and a CAVE construction.

A HMD is worn by a user. Through the displays of the HMD the user sees the virtual environment. A tracking system is needed to follow the users’ head-movement in order to show the right perspective of the virtual environment. An HMD is made for individual use, but the real-world can be totally shut out. That last thing is slightly different with the CAVE (Computer Automatic Virtual Environment) that consists of a cubic like structure in which the participant (and the therapist) can take place. The virtual environment is projected on the four to six sides of the cubicle. The participant wears shutter glasses. These shutter glasses lighten and darken in synchronization with the images on the screens. This way the left and the right eye see the same image from a different perspective that corresponds with the positions of the eyes. This way they see the projections on the screens in 3D. Also a tracking system keeps track of the position of the shutter glasses. A sensor is attached to the patients shutter glasses to generate a correct perspective view. The main difference with HMD is that the CAVE can be used with multiple users at once, while the HMD is made only for single usage. Then again when using the HMD the real world can be totally shut out, which isn’t the case when standing in a CAVE. Study also showed that the effectiveness of treatment of fear of heights using a HMD is the same as using a CAVE, while HMDs are much cheaper and easier to use than CAVEs.[15]
Besides the HMD and the CAVE there are also other techniques. For example Tactile feedback is used to add another stimuli to the user experience, namely a tactile stimuli. This can be accomplished using a Hand-sensing glove. This is a glove with vibro-tactile stimulators on each finger and palm. Hand sensing gloves make it possible to interact with the environment using your hand. Force feedback systems also produce tactile stimuli. Force feedback systems for your hands, make it possible to touch and hold objects in the virtual environment and feel a (force) resistance as you hold the object. The tactile feedback chair is a migration of the tactile feedback from other areas to a chair. The chair can give tactile feedback to the user in the form of vibration. This way the feeling of presence of the participant can be increased. For example, when simulating sitting in an airplane, the vibrations that the airplane produces can also be simulated.

### 3.2 Presence

Presence is an important concept in VR, but still there are a lot of theories regarding the nature of presence. This is mainly because presence is a rather subjective concept. Presence can be described as the sense of “being there”, especially when talking about a virtual environment. All these theories do not necessarily contradict each other, although they can be quite different. They all share some similarities. One common factor is that of attention. In fact ‘Presence’ and ‘Situation awareness’ are overlapping constructs. Many authors have assumed a strong relation between presence and the level of interactivity [23]. Below we will shed light on a few theories regarding presence.

#### 3.2.1 Theories

**Subjective and Objective presence**

Schuemie [24] discusses several theories regarding presence in his article on Presence in Virtual Reality. One of the discussed theories is that of Schloerb. Schloerb distinguishes two types of presence:

- **Subjective presence**, the likelihood that the person judges himself to be physically present in the remote or virtual environment; and
- **Objective presence**, the likelihood of successfully completing a task.
This definition however is completely empirical and isn’t one of the most commonly used theories regarding presence.

**Exclusive presence**

Another theory is **Exclusive presence** by Slater [30]. He stresses that the participant’s sense of “being there” in the virtual environment, and point out that a high sense of presence in a VE requires a simultaneous low level of presence in the real world and vice versa. So in this case it is only relevant to talk about the degree of presence in one environment. In figure 3.3 we see the visual presentation of this theory.

![Figure 3.3. Exclusive presence: relationship between Virtual- and Real world presence](image)

**General theory of presence (Mental models)**

This is the most general theory regarding presence. When interacting with a VE, we can distinguish two different mental models [24] as can be seen in figure 3.4:

1. The model of the Real World (RW)
2. The model of the Virtual World (VW)

![Figure 3.4. General theory of presence](image)
In this theory presence refers to the distinction made by the user between the RW and the VW. Like exclusive presence these two models overlap and being more present in one, means being less present in the other. The Virtual as well as the Real World can be divided into the ‘Self’, which is the mental model that the person has of him or herself. The ‘Non-self’ is the environment as the individual experiences it. The non-self can even be further divided into a social model and an environment model. We now have a tree-like structure, where each leaf represents a specific type of presence. Heeter[11].

- **Personal presence** is related to the ‘Self’. It is the measure of extent to which a person feels like he is really in the virtual environment.
- **Social presence** relates to the social model. Commonly this type of presence is described as the extent to which a medium is perceived as sociable, warm, sensitive, personal or intimate when it is used to interact with other people. Social presence can also be achieved using synthetic beings (e.g., A creature that gives tips, talk with you or keeps coming back).
- **Environmental presence** refers to the environmental model and indicates the extent to which the environment reacts on the person or seems to know that the person is there.

### 3.2.2 Measurement of Presence

Because presence is a rather subjective concept, it is also hard to measure. We can however distinguish two types of measurements.

**Subjective measures: Questionnaires**

This type of measurement is most commonly used. The patients give subjective ratings through questionnaires. These questionnaires can be one or more questions on all sorts of variable themes. For example:

- The subject’s sense of “being there”;
- The extent to which the VE becomes more “real or present” than everyday reality;
- The “locality,” the extent to which the VE is thought of as a “place” that was visited rather than just a set of images.

**Objective measures: Behavioral**

This type of measurement is done by studying the patients’ behavior as a response to mediated stimuli (e.g. heart rate, skin temperature, or skin conductance). But this type of measurement is rather tricky because of the noise in the signal. A increasing heart rate could also be caused by other factors than the stimuli from the virtual world. The heart rate could have increased because of arousal instead of feeling presence. And also the results are hardly comparable, because of user characteristics. Humans can have different behavioral reactions to the same situation.
3.2.3 Causes of Presence

Much research has been devoted to finding factors that contribute to presence. In the article of Schuemie [23], we see that several researchers have already made some categorizations of these factors. Following we give an enumeration of these factors.

a. Slater and Usoh [30]
   - High quality, high resolution information
   - Consistency across all displays
   - Interaction with environment
   - Virtual body, the representation of the users’ body in the VE.
   - Effect of action should be anticipated

b. Witmer and Singer [45]
   - Control factors: degree of control, immediacy of control, anticipation of events, mode of control, physical environment modifiability.
   - Sensory factors: sensory modality, environment richness, multimodal presentation, consistency of multimodal information, degree of movement perception, active search.
   - Distraction factors: isolation, selective attention, interface awareness
   - Realism factors: scene realism, information

c. Sheridan
   - Extent of sensory information
   - Control of relation of sensors to environment.
   - Ability to modify physical environment

d. Steuer
   - Vividness refers to the ability of a technology to produce a sensorially rich mediated environment.
   - Interactivity refers to the degree to which users of a medium can influence the form or content of the mediated environment.
   - User characteristics refers to the individual differences in users.

e. Lombard and Ditton
   - The form in which the information is presented.
   - The content of the information.
   - User characteristics.

3.3 Anxiety Disorders

Anxiety is one of the many human emotions. It is a normal reaction to stress. It helps to deal with a tense situation at work, keep focused on an important speech or workload. In general, it helps one cope. But when anxiety becomes an excessive, irrational dread of everyday situations, it has become a disabling disorder.

According to DSM-II-R (Diagnostic and Statistical Manual of Mental Disorders) [6], Anxiety disorders can be divided into the following categories:
DSM-III-R categories of anxiety disorders

- Panic disorder with agoraphobia
- Panic disorder without agoraphobia
- Agoraphobia without history of panic disorder
- Social phobia
- Obsessive-compulsive disorder
- Post-traumatic stress disorder
- Generalized anxiety disorder
- Anxiety disorder not otherwise specified

In this document we focus on agoraphobia with panic disorder. The first thing we must understand is that there are two kinds of agoraphobia: agoraphobia with panic disorder and agoraphobia without panic disorder. The term agoraphobia has widely been misunderstood. It literally means fear of “open spaces”, but in reality it means more than just that. Before we go deeper into agoraphobia with panic disorder, we first focus on what panic disorder exactly means.

3.3.1 Panic disorder

Someone with a panic disorder has periodic panic attacks. A panic attack is an unexpected period of intense fear or discomfort [33]. A person has to experience at least four attacks in a period of 4 weeks to be diagnosed with panic disorder. During at least one of the attacks four of the following symptoms occur within 10 minutes of the beginning of the first symptom with which the attack commences [6].

1. Shortness of breath (dyspnea) or smothering sensations
2. Dizziness, unsteady feelings, or faintness
3. Palpitations or accelerated heart rate (tachycardia)
4. Trembling or shaking
5. Sweating
6. Choking
7. Nausea or abdominal distress
8. Depersonalization or derealization
9. Numbness or tingling sensations (paresthesias)
10. (Hot) flushes or chills
11. Chest pain or discomfort
12. Fear of dying
13. Fear of going crazy or of doing something uncontrolled

These panic related symptoms however, can also be caused by other factors. For example restlessness, nervousness and increase heart-beat can also be caused by excessive use of coffee or other caffeine-containing products. This makes it difficult to determine whether certain symptoms are a result of a panic attack or not.

Persons that suffer from panic disorder are actually afraid of getting into the feared situation of losing control. So in the moment of anticipation they already feel an increased arousal.

We may distinguish four types of loss of control.
1. Fear of somatic loss of control (fearing a heart attack, a stroke or fainting)
2. Fear of Psychic loss of control (fear of going mad or not being able to think properly anymore)
3. Fear of behavioral loss of control (fear of losing control over their own behavior)
4. Fear of social loss of control (fear of losing social respect because of signs of increased arousal such as nervousness, trembling, wanting to leave)
3.3.2 Agoraphobia

Agoraphobia with panic disorder

Agoraphobia with panic disorder is the most common form of agoraphobia. It happens when a person that suffers from panic disorder is afraid of being in places or situations from which escape might be difficult, or embarrassing, or in which help might not be available in the event of a panic attack. For example a person that has had several panic attacks in a crowded subway, can avoid the subway in order to prevent the panic attack. In the worst case a person might stay housebound, because he or she has had several panic attacks in several places (like the elevator, a crowded square or a subway) and avoids all of these places and situations [4]. So in fact panic can be seen as the precursor of agoraphobia. Agoraphobia with panic disorder is a combination of panic attacks and avoidance behavior. The core element of agoraphobia is “the fear of the fear”.

Agoraphobia without panic disorder

The difference with “agoraphobia with panic disorder” is the motive of the avoidance. In the first case persons tend to avoid panic attacks and stay out of situations from which it is hard to escape or where there is no help at hand in case of a panic attack. But in the case of “Agoraphobia without panic disorder” there where no preceding panic attacks. These persons have the fear of suddenly emerging symptoms which may cause embarrassment to the person or make him or her in need of help. The most common fear is to lose control over the bladder or bowels, to have to vomit, depersonalization or derealization, and dizziness [6].

Gradations of agoraphobia

According to the Diagnostic and Statistical Manual of Mental disorders (DSM-III-R) there are gradations of agoraphobia [6]. Mild agoraphobia means that a person has some avoidance, but can live a relatively normal life. The person is able to leave the house for necessary activities, but does not consider further traveling alone. In the case of Moderate agoraphobia the person is afraid of being more than a few kilometers away from home without company. These persons have a relatively restricted life style. Severe agoraphobia is the highest gradation of agoraphobia. In this case the person is totally housebound and hardly dares to leave his or her home.

Anxiety provoking situations

One of the most characteristic features of agoraphobia is the avoidance aspect. In the case of Agoraphobia with a history of panic attacks, the number of situations that cause anxiety are as diverse as the number of persons that suffer from agoraphobia, because it’s depends on the persons panic attack history. Agoraphobia is a fear of fear and not so much a fear of certain places. But the central theme is “not being able to leave” or “being stuck”. So for patients to feel present in the virtual world (or place) and really experience anxiety the patients have to get the feeling that they are not able to leave or escape the situation. Of course this is hard to do since the patient can take of the Head Mounted Display at any time. But the fact that the virtual world is imitating a place or situation from which escape is hard or impossible, will cause a sufficient level of fear for the agoraphobic.
Another aspect of Agoraphobia that has to be taken into account is that most agoraphobics are much more fearful when alone and not accompanied by a trusted person. So they often avoid being alone.

A few example situations are as follows:
- Standing in a queue
- Being in a large shop or shopping center
- Traveling by public transport (bus, train or airplane)
- Crowds, busy streets, large gatherings
- Driving a car on a motorway (the impossibility of turning on the road)
- Being in a traffic jam
- Crossing a bridge or being on a bridge
- Sitting at the barber’s
- Being in conversation with some person on the street

3.4. VRET

3.4.1 Traditional treatment

Different methods are used in order to treat patients with certain phobias. Cognitive Behavioural therapy is one of the most common methods. It is a combination of Cognitive Therapy and Behaviour therapy. In cognitive-behavioural therapy, therapists recognise the maladaptive cognition and replace them with adaptive cognition [6]. It is based on the scientific fact that our thoughts cause our feelings and behaviours, not external things, like people, situations, and events. The benefit of this is the fact that we can change the way we think to feel / act better even if the situation does not change. Patients utilise imagery of anxiety-provoking situations to identify unrealistic thoughts, challenge these thoughts, and substitute more adaptive ones in their place.

Exposure

An important aspect of this kind of treatment is exposure. When people that suffer from a phobia are gradually exposed to the anxiety provoking situation, it helps them to create more neutral memory structures that ‘overrule’ the old anxiety provoking ones. We make a distinction between exposure in vivo and exposure in vitro. Exposure in vivo means that the phobia treatment is done in real life with real life situations. Exposure in vitro means that the treatment is done with the patient imagining the anxiety provoking situation. This technique leans on the imagination of the patients, which is different for each patient. Research showed that exposure in vivo is more effective in treating phobias, than exposure in vitro. A drawback of exposure in vivo is, that some patients might find these real life situations ‘too’ provoking and refuse treatment.

External and interoceptive exposure

According to Botella [5], panic disorder and agoraphobia sufferers usually avoid two different kinds of stimuli. External and interoceptive stimuli.

External exposure comes from being in the feared situation (e.g., When a person that suffers from claustrophobia is put in an elevator that person gets external stimuli and thus is exposed to a situation that it usually avoids)[3].
Interoceptive exposure consists of exposing the patients to the feared bodily sensations, similar to the ones experienced in panic attacks. (e.g., hyperventilation, blowing through a straw, running, jumping). In order to have more therapeutic benefits, interoceptive and situational exposure should be conducted at the same time. So in terms of VRET, not only should the patient be exposed to the world, but also to as much bodily sensations as possible (e.g., breathing difficulties, increasing heart rate, tunnel vision, blurred vision).

### 3.4.2 Treatment with VRET

Virtual reality (VR) is a technique that is progressing a lot, because the numerous research projects that have been done. It is still in it’s infancy, but we can already say that future prospects are very positive. Recently there has been a plenary discussion at the NATO Advanced Research Workshop on Novel approaches to the diagnosis and treatment of posttraumatic stress disorder [18]. Basic functions for a VRET system were provided and the promising future of Tele-care were discussed. The possibilities of application are also growing. On of the application fields of VR is the field of phobia treatment. Because we can construct virtual environments with this technique, it is interesting to build, but we can already say that future prospects are very positive. The possibilities of application are also growing. On of the application fields of VR is the field of phobia treatment. We can treat patients by exposing them to a virtual environment, instead of a real life environment. This is called Virtual reality exposure therapy (VRET). VRET actually is the process of desensitization, for people with phobias, using virtual reality technology. It is quite like exposure in vivo, only the real life situation is a virtual computer generated environment. We can even say that it could be placed between exposure in vivo and exposure in vitro. The more realistic the virtual environment is the more it tend to be the same a exposure in vivo. VRET is still in an experimental stage. As the article of Foa and Kozak (1986) states, there are three conditions that should be met for VRET to be effective. First, patients need to feel present in the virtual environment. They need to have the feeling that they are really ‘in’ the virtual environment. Second, the virtual environment should be able to elicit emotions (e.g. anxiety). Third, extinction and co-occurring cognitive changes have to generalize to real situations so that real-life situations will not be avoided any longer or will be endured with less anxiety. Virtual Reality Exposure Therapy could be a interesting alternative for exposure in vivo, because it has many advantages.

- **Time / Costs:** VR treatment can really help to overcome some of the limitations of one of the main therapeutic components to treat this problem and that is exposure. Exposure in vivo is more expensive and time consuming. Imagine what it would take to treat a person with fear of flying, by actually taking real flights or to take a patient with fear of heights to the top of a building.
- **Control:** Because the environment is virtual the therapist has high accuracy control over the environment.
- **Repetition:** All situations can be repeated and for as long as needed.
- **Scenarios:** The number of scenario’s that can be created is unlimited, due to the fact that the situation is virtual.
- **Safety:** Because the environment is virtual and controlled, this gives more safety to the patient and the therapist.
4 Current system

4.1 Introduction

The Delft, Technical University and the University of Amsterdam developed a VRET system that can be used for treating patients with several phobias, including fear of flying, claustrophobia, agoraphobia and fear of heights. This current system has been in use for the past five years. The current system has been successfully used in the treatment of several phobias.

4.2 Techniques

The current system uses the following soft- and hardware components:

**Hardware components** Schuemie [26]

<table>
<thead>
<tr>
<th>INPUT</th>
<th>Tracker system</th>
<th>Ascension Flock of Birds with a transmitter and control unit. Connected to the computer using serial RS-232.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical control</td>
<td>Joystick</td>
<td></td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Visual device</td>
<td>HMD Visette Pro (Cybermind) Stereoscopic, 70 degrees FoV, Resolution: 640x480</td>
</tr>
<tr>
<td></td>
<td>Haptic device</td>
<td>Aircraft chairs with built-in woofers</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>2 Pc’s with Pentium IV technology (2,66 GHz) with one nVideoQuadro4 videocard and one standard 3D videocard.</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.1 – Hardware components for the VRET system at the TU Delft

**Software components**

The software that was used to develop the virtual environments is WorldUp R4 by Immersion and Borland Delphi 5 is used for additional functionality. World Up has got a build-in modeller, but it is recommended to use an external modeller such as Maya or 3DStudio Max.
Virtual Environments

The current system uses the following environments:
§ Mall in Amsterdam (Acrophobia)
§ Fire escape (Acrophobia)
§ Roof garden (Acrophobia)
§ Lift (Claustrophobia)
§ Small path (Claustrophobia)
§ Airplane (Fear of Flying) figure 4.2 and 4.3
§ Subway cabin (Agoraphobia)
§ Square (Agoraphobia [in progress])
§ Schiphol Airport (Fear of Flying)

Figure 4.2 – Screenshot of the Airplane World for “fear of flying”

Figure 4.3 The therapist UI of the Airplane World for “fear of flying”
4.3 Current System Framework

In figure 4.4 we see an overview of the current framework [21] and how the previous software and hardware components are integrated.

The system uses two computers. One for the therapist user interface (therapist computer) and one for the patient user interface (patient computer). The patients’ user interface consists of a stereoscopic view, which is send to the stereoscopic HMD. Position and orientation of the HMD is detected by the Flock of Birds (Ascension). The therapists’ user interface contains two viewpoints: One is the same viewpoint as the patient, the later is an additional free viewpoint. This second viewpoint is used to solve navigation problems experienced by some therapists [28]. The virtual environments, made with WorldUp, are locally run on both computers. For each virtual environment, another world (application) has to be executed. The interaction between the two computers is established by a socket connection. This is achieved by integrating DLL-functionality in the VR-environments. These DLLs were written in Delphi. Information regarding the location and orientation is sent from the patient computer to the therapist computer. The therapist can make changes in the virtual environment by triggering events (e.g. turning of the light, closing and opening doors etc.) These world parameters are sent from the therapist computer to the patient computer. This data is sent ten times a second in both directions.

Figure 4.4 – Overview of the current system [21]
4.4 Migrating the VRET system to Windows XP

The initial version of the system was running on Windows 98 and Windows NT platforms. Afterwards, a conversion was made to Windows 2000. Even with this conversion the system was still outdated in both software and hardware. In order for the system to be used in the future it had to be migrated to the Windows XP platform. Together with S. Roorda [21] we migrated the system to Windows XP. This was not a straightforward conversion. It was not possible to simply copy the files to the new platform and run it. In fact, there were a lot of problems during this conversion, which took several months to finish. The process was documented by Roorda in an internal report [22]. The main problems were caused by the incompatibility of the application with the hardware on the new platform. The higher performance of the new platform caused problems, because the old system was not prepared for such a high performance.

There was hardly any documentation of the old system, so before we could do the migration we had to analyse the whole system to get a thorough understanding of the system. Since the first implementation of the system by M. Schuemie, several students have worked on the project in order to adjust, update or extend the system. Some of the changes that were made during these years, were not documented. This made it also complicated to get a thorough understanding of the system, because some of the elements that Schuemie described in his PhD thesis, were not up to date anymore [28].

4.5 Conclusion

After the migration to Windows XP the current system is quite up to date. The hardware may get out of date in a couple of years, but this is not completely necessary. As we stated previously the main concern was the discontinuation of the software manufacturer of World Up (Sense8). This is the main reason for a proposed new framework. Roorda [21] has done research on the candidate applications to replace World Up. A new application brings new properties and possibilities and thus causes a change in the current framework. Some improvements can also be made, regarding the overall framework. In the current system each virtual environment has to be launched separately. It would increase the usability of the system, if the user (therapist) could switch between environments, without first having to shut down the application and run a new one. In the optimal case the different virtual environment are interconnected. For example: One can move from a square to take the subway, that brings you to the airport, where a flight can be taken. Another improvement would be the structuring of the files. The current system consists of a collection of different files on each computer. In order to make the system work these files have to be placed manually in directories and subdirectories on different hard drives. It would be an improvement if the files could reside in one directory or in the optimal case could automatically be placed in a proper destination using an installer.
5 Proposed system

5.1 Introduction

In this chapter we will give an impulse towards a new framework. We do this by designing and implementing a virtual environment that can be used for VRET for agoraphobia. It is not our goal to come up with a complete new framework, but this virtual environment serves more as a prototype for worlds that can be build in the future using a new framework and thus gives a first step towards this new framework. We choose to implement a virtual environment simulating a well known square in the Netherlands: The Market in Delft. We choose the Market for its large surface, historic background, international identity and its diversity. Primarily the large surface and diversity create a lot of opportunities for this research.

5.2 Requirements analysis

5.2.1. Parameters

During our research on the parameters that are necessary for creating valid and anxiety-provoking virtual environments for the treatment of agoraphobia, we came up with the following conclusion [12]:

The necessary parameters for creating valid and anxiety-provoking virtual environments for the treatment of agoraphobia can be divided into two groups: One common group of presence parameters that contribute to the feel of presence, which are suitable for any phobia. For these parameters we used the general theory of presence. All of the five parameters constitute to one of the three forms of presence. A second group of anxiety provoking parameters contributes to the level of fear that the patient experiences. This last group of parameters is phobia specific. Also this last group can have a subdivision into their contribution to interoceptive- and external stimuli.

We also concluded that these two types are closely related. For example when the feel of presence is high, but the situation (in the virtual environment) is not provoking any anxiety the effectiveness is low. (E.g. Someone with fear of heights that is put in a realistic virtual environment for claustrophobics where he or she feels really present.) Also when the virtual environment is a provoking situation, but the patient does not feel present, the effectiveness is also low.

For the exhaustiveness we give a summation of the two parameter groups.

A. Presence parameters

- Level of realism
  Of course the more realistic the virtual world is, the more the feel of presence will increase. However, patients who suffer from phobia, do not need hundred percent of realism in order to provoke anxiety. When we look at former VRET projects, a resolution of 640x480 has seemed to give a sufficient level of realism. Another aspect of the level of realism is the Field of View (FoV). The FoV stands for the area that is visible, as through a HMD. This area is measured by an angle. The larger the angle, the larger the field of view, thus the larger the level of realism.
• **Virtual body**
  A representation of a user’s body in the virtual environment, contributes to the feel of personal presence.

• **Number of sensorial modalities**
  Sensorial modalities are visual, auditory, tactile (feel, touch) and olfactory (sense of smell) The more sensorial modalities the system covers, the more the user feels present in the virtual environment.

• **The level of interaction with and existing of other creatures in the virtual environment.**
  The more the patient is able to interact with other (virtual) people in the virtual environment, the more he/she will feel present in the environment. Avatars with expressions and inferred-gaze could have a positive impact on perceptions of communication and thus on the feel of being present [8].

• **Level of interaction with the environment**
  This accounts for the degree to which users of the medium can influence the form or content of the mediated environment. We can measure this by looking at three aspects. 1. Speed - The speed with which the medium responds to user inputs. 2. Range – Amongst others the number of inputs from the user that the medium accepts and to which it responds. 3. Mapping – The degree of correspondence between the type of user input and the type of medium response. Examples of user inputs are mouse, joystick, keyboard, hand-sensing glove, tracker (e.g. tracking head movement).

**B. Anxiety provoking parameters**

These parameters contribute to the level of fear experienced by the patient. According to Botella [5], panic disorder and agoraphobia sufferers usually avoid two different kinds of stimuli. External and interoceptive stimuli. So parameters that contribute to the level of fear can also be subdivided into two types. Simultaneously conducting these parameters will show best results.

**B1: Parameters that contribute to External stimuli**

• **Level of possible escape**
  This parameter accounts for the situation that is simulated by the virtual world. Agoraphobics fear situations from which escape is hard or impossible. So this parameter should be kept as low as possible. Below follows a list of situations where the level of possible escape is very low as we saw earlier on.

  $\checkmark$ Standing in a queue
  $\checkmark$ Being in a large shop or shopping center
  $\checkmark$ Traveling by public transport (bus, train or aeroplane)
  $\checkmark$ Crowds, busy streets, large gatherings
  $\checkmark$ Driving a car on a motorway (the impossibility of turning on the road)
  $\checkmark$ Being in a traffic jam
  $\checkmark$ Crossing a bridge or being on a bridge
  $\checkmark$ Sitting at the barber’s
  $\checkmark$ Being in conversation with some person on the street
• Level of habituation

As we saw in the previous chapter, by no means may the patient get habituated to the environment, because this causes a decrease in the patients level of fear. The level of habituation must be kept as low as possible. This can be achieved with a virtual environment has as much variation as possible.

B2: Parameters that contribute to Interoceptive stimuli (Bodily sensations)

• Number of (agoraphobia specific) bodily sensations

This is the number of agoraphobia specific bodily sensations that can be simulated with the system. Agoraphobia specific bodily sensations that can be simulated are [6]:

- Increasing/decreasing heart-rate
- Blurred vision
- Tunnel vision
- (Hot) flushes or chills
- Sweating
- Choking
- Trembling or shaking
- Shortness of breath (dyspnea) or smothering sensations

5.2.2. Requirements

The above parameters have to be taken into account for the implementation of the world. However a trade-off has to be made between the effectiveness and costs. From the above parameters we can easily fill in some of the requirements for our virtual environment that will be build.

First we take a look at the presence parameters, because these account for any virtual world. The level of realism is sufficient when we use a resolution of 640x480 [31]. The HMD that is going to be used for viewing our virtual environment has this specification, thus we automatically account to this parameter.

A virtual body is also a parameter that contributes to the feel of presence. We choose not to implement a virtual body, because in this case it can even work backwards. The participant is really mobile. He or she is not just sitting on a chair, but is actually navigating through the virtual world using his or her legs. To implement a valid virtual body, we would have to map the movement of the legs (and maybe arms) onto the virtual environment. The overhead that this costs, does not account for the gain in feel of presence [32].

We choose to use two sensorial modalities, namely visual- and auditory modalities. We choose to use these two modalities and leave the tactile and olfactory modalities behind, because they contribute the most to the feel of presence [24]. The presence gain of tactile and olfactory modalities is still being researched and techniques to implement these modalities into a virtual world are still in its infancy.

The existence of other creatures in the virtual environment is an important aspect for people who suffer from agoraphobia. Even though the level of interaction is not high, the fact that there exist other creatures causes anxiety [8].

The level of interaction must also be high, so the user must have six degrees of freedom, even though the therapist controls some of these freedoms.

Secondly we take a look at the anxiety-provoking parameters to extract requirements for the new system. The level of possible escape shows a strong relation with the situation that the virtual environment simulates. In order to keep this parameter as low as possible we choose one of the situations from the list of
anxiety provoking situations by Emmelkamp: “Crowds, busy streets, large gatherings”. This situation is realized in the form of a large square, with a crowd on it.

The level of habituation must be as low as possible. This can be done by making the world heavily adaptable. Examples for our case are the changing of weather, day and night, the textures of the houses or the number of people on the square. This way the participant will not get the chance to habituate to the environment. Also using different environments for the same situation is a suitable option.

From the list of bodily sensations, we decided to choose only the ones that could be simulated with auditory- and visual modalities. Thus we have the following bodily sensations: Increasing/decreasing heartrate, Shortness of breath, Blurred vision and Tunnel vision. The first two bodily sensations however, are simulated with auditory modalities and are not the actual heartrate and breath of the participant. In fact they are used to fool the patient. According to Emmelkamp [7] the sound of an increasing heartrate, causes the participant to have an increasing heartrate. Also the sound of a decreasing heartrate has the opposite effect. This is the same for the breathing.

So before we go to the task analysis, for the completeness we will give a summation of the overall requirements.

Common requirements
- Sufficient resolution 640x480
- Visual- and auditory sensorial modalities
- Existence of other creatures in the environment.
- High level of interaction (6DoF)

Phobia specific requirements
- Simulation of a Phobic situation (in this case a square which is a feared situation for agoraphobics)
- Heavily adaptive virtual environment (for less habituation)
- Simulation of bodily sensations

5.3 Task analysis

During our research on the parameters that are necessary for creating valid and anxiety-provoking virtual environments for the treatment of agoraphobia, we interviewed prof. Emmelkamp. After this interview we came up with the following task analysis, that was approved by Emmelkamp.

A. The therapist must be able to navigate the patient through the virtual world
B. The therapist must be able to make the patient believe that the patients heart-rate is higher or lower than it actually is. This way the therapist can manipulate the patients feeling of fear for the virtual world. Because when a patient becomes frightened his or her heart-rate will go up resulting in a catastrophic interpretation that even causes more anxiety and a higher heart-rate. When the therapist however makes the patient believe that his or her heart-rate is not increasing at all, the patient will calm down. This method could also be applied for other Agoraphobia specific bodily sensations like hyperventilation, shortness of breath etc.
C. The therapist should also be able to make the ambience noise more intense or less intense. (E.g. the sound of talking people and cars).
D. The therapist should also be able to change the weather in the Virtual Environment (Rain, Thunder, Sunshine etc.)
E. The therapist should also be able to choose between different types of squares. The patient may by no means get used to a certain environment, because this will decrease the level of anxiety.
F. The therapist should be able to change the size of the square.
G. The therapist should be able to influence the patients sight, applying ‘blurred vision’ and/or ‘tunnel vision’. These are agoraphobia specific bodily sensations, and will increase the anxiety level of the patient.
H. The therapist should be able to change the number of people on the square. For example the therapist could have a choice between ‘empty’, ‘calm’ and ‘crowded’. The transitions between the different types of occupations should be animated. So when changing from ‘empty’ to ‘crowded’, people will gradually crowd the square. The most optimal situation is when people who are on the square, move, react and act in a realistic manner.

I. Except for the above parameters there were also some other requirements that do not directly constitute to the anxiety provoking elements of the world, but are merely used as a support of the therapist. These are the informational requirements which are summed up below, will not be implemented in the prototype but serve more as handles that can later be used for the future framework.

J. The therapist should be able to determine the patients fear, using Selective Units of Discomfort (SUD). The therapist asks the patient to report their level of fear. Often the patient is instructed to use a scale from one to ten to report their fear. This is called a SUD.

K. The therapist should be able to store sessions in a database, where patients have a patient number and one patient has multiple sessions stored. A session consists of an array of SUDs given by the patient during that session and additional comment from the therapist. The therapist can then log in to the system using the patient number and view the session history of a patient or start a new session.

L. The therapist should have a free viewpoint to keep track of the patients position. The images generated by the system show only one viewpoint, the viewpoint of the patient. This viewpoint is essentially used to expose the patient to the fearful stimuli and shows only a small part of the VE at one time. When the therapist tries to use this view to move the patient through the VE, some obstacles in the VE might not be visible, making navigation difficult, especially in small spaces. The free viewpoint is by default locked on the patient, showing the viewpoint of the patient. And the free viewpoint can rotate and translate around the patient, solving the navigation problem of some therapists.

5.4 Techniques

As we previously stated, S. Roorda [21] has done research on the candidate applications to replace World Up. Amongst several applications Quest3D became the most suitable software package. Roorda based his conclusion on a point system, with a range from one to five, in which Quest3D scored the highest points on almost each area. The seven areas were: Hardware support, software development, scene development, conversion and import, price, licensing conditions and continuation. This can be seen in the associated table 5.1. In this paragraph we are going to explain the global concept of Quest3D and our experiences using this application. For this implementation we used the following version of Quest3D: Quest3D 3.0d Educational Edition.

<table>
<thead>
<tr>
<th></th>
<th>CAVElib</th>
<th>CaveUT</th>
<th>EON Reality</th>
<th>MetaVR</th>
<th>MultiGen</th>
<th>Quest3D</th>
<th>Virtools</th>
<th>Vizard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware support</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Software development</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Scene development</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Conversion and import</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Price</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Licensing conditions</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Continuation</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>21</strong></td>
<td><strong>29</strong></td>
<td><strong>19</strong></td>
<td><strong>19</strong></td>
<td><strong>23</strong></td>
<td><strong>31</strong></td>
<td><strong>29</strong></td>
<td><strong>31</strong></td>
</tr>
</tbody>
</table>

Table 5.1 Comparison between Quest3D and other candidate applications [21]
5.4.1 Quest 3D

Quest3D is an extensive software package for creating interactive 3D scenes [38]. It features a unique style of programming. Instead of having to write thousands of lines of complex code, developers make use of a large set of powerful building blocks, both flexible and easy to use. The working environment of Quest3D is in real-time. So you are working directly on the end result and any change you make in the environment is directly visible.

The interface consists of three main sections [1]:

- The Channels Section: This is the heart of Quest3D. Quest3D shows the Channels Section on startup. In this section you can build application by creating a network of interconnected channels. In the Channels Section you will build the foundation of all projects to create.

- The Animation Section: This is where 3D objects, camera’s and lights are positioned and animated. It contains a large preview window perfect for testing projects.

- The Object Section: This is where 3D objects are prepared and polished. Quest3D provides a wide range of options to tweak object surface properties such as color and texture.

The elementary building blocks of Quest3D are channels. In figure 5.1 we see an example of a channel. A channel has an input and output and some functionality. Information and parameters regarding this functionality can be seen in the channel properties window, which we see on the right. The small black squares above and beneath a channel are called link squares. Channels can be connected to each other by lines between top and bottom link squares.

![Example of a channel and a channel properties window](image)

Figure 5.1 – Example of a channel and a channel properties window

There is a large collection of different channels, all with their own functionality. For example we have a value channel, which simply holds a value or an expression channel which take the input of one or more value channels and execute an expression on these values. There are also channels that handle sound, networking or databases. Channels are connected to each other in a way that they produce some overall
functionality. A group of interconnected channels is called a channel group. In Quest3D there are also templates. A template is a predefined channel or group of channels. Using templates can greatly speed up your workflow. Templates can be added just like single channels. In figure 5.2 we see an example of the Vector template.

![Figure 5.2 – Example of a vector template](image)

The program flow of Quest3D is realized as a tree-like structure. A Quest3D program begins with a Start Channel. A Start Channel is marked by a large arrow pointing down. Quest3D works in real-time. This means it constantly executes a project completely and updates the preview. One complete cycle through the channel structure is called a frame. Calculating all end results is referred to as rendering. Calling a channel in Quest3D means carrying out all of its functionality once, based on any input it may receive from its children. A Channel Caller calls its children in a left to right order.

3D objects play an important part in most Quest3D programs. 3D objects can be imported from an external 3D modeling package like Maya or 3D Studio Max. (In this project however we choose Maya). You can also use one of the 3D objects that come with Quest3D. Just like a movie-set, a Quest3D scene needs a camera and one or more lights. The Render channel is used to display 3D objects on the screen. In figure 5.3 we see an example of a scene. The ‘Start 3D scene’ is used to initialize the 3D environment. As you can see the Camera, 3D Object and Light channel have multiple link squares. Multiple children channels can be connected to these link squares to determine the functionality, parameters, look etc. of the channel.
When you have finished your Quest3D application you can publish it to either a project file (which can be viewed on any computer that has a Quest3D viewer installed) or an executable that does not need any viewer and can directly be executed.

### 5.4.2 Evaluation of Quest 3D

For our implementation, we had to learn and evaluate Quest3D. We must admit that there is a long learning curve, because the concept of Quest3D differs much from a regular programming environment. Below we sum up our experiences during the use of Quest3D.

**Foreknowledge**

Quest3D expects the user to have a lot of foreknowledge on general 3D graphics theory. Subjects like emission, ambient light, diffusiveness, material/shading, z-buffers, collision detection are expected to be known subjects to the user. Also a lot of linear algebra knowledge, for matrix calculations is expected to be present knowledge. Thus before we could even start using the package we had to read in a lot on 3D graphics theory in general. This also accounted for the relative long learning curve.

**Complexity Explorer**

A simple virtual environment, with a few functionalities can quickly become a complex network of interconnected channels in folders and subfolders. It is a disadvantage of Quest3D that there is no tool, to have some overview in this complexity. There is a search function, but this is quite basic. A general explorer would be very useful.
Debugging

Quest3D has no real debugging function. When something unexpected occurs in your virtual environment, you have to track down the channel in a complex network of interconnected channels in folders and subfolders.

Textures

An important thing to keep in mind when using Quest3D is the effect of large textures. Do not use textures larger than 1024x1024. Using large textures can slow up your virtual environment dramatically. We did a test with two textures of a different size. One texture of 1024x1024 and one of 256x256. We used the textures that were holding the same image, to texturize the sky. Next we used a virtual environment that already has a low performance and tried to navigate through the world using a Walkthrough camera. With the small texture there were no problems at all, but with the large texture the navigation stammered. The key rule is not to use a resolution for a texture that is bigger than it is going to be on the screen [1].

Publishing

Quest3D is a real-time development environment. However there is a large performance difference between a virtual world in the development environment and the published environment. We did measurements with a virtual world, where we tried to navigate from one part of the world to the other part. The world consisted of a simple square with five buildings and a sky sphere. We measured the time (in seconds) it took to walk a straight path from one part of the world to another. The path has a fixed distance of 40 units by moving the camera from position (-23, 1.4, 9) to (-23, 1.4, 49). All the measurements where done on the same computer (Pentium IV technology (2.66 GHz) with a standard windows XP compatible 3D videocard).We also did these measurements using different resolutions for the world in the publishing as well as in the developing environment. In the table below and in figure 5.4 we see the results of these measurements.

<table>
<thead>
<tr>
<th>Environment</th>
<th>Resolution</th>
<th>640x480</th>
<th>800x600</th>
<th>1024x768</th>
<th>1280x960</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development</td>
<td>21.67s</td>
<td>24.43s</td>
<td>27.02s</td>
<td>29.54s</td>
<td></td>
</tr>
<tr>
<td>Published</td>
<td>20.84s</td>
<td>22.12s</td>
<td>24.89s</td>
<td>26.08s</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2 Time measurements of walking a path with different environments and resolutions.
In the above table we see that there is a difference between the performance in the development environment and the published environment. However the biggest difference comes from the resolution changes. In our case we use the resolution of 640x480. However when we are developing in Quest3D we are using a more standard resolution of 1024x768 or 1280x960. So this tells us that during the development process we have to take into account that the performance of our application will be seemingly better when we publish the project and use a lower resolution. We recommend to test a published version of the application now and then, under the low resolution so you can estimate the eventual performance. This performance difference had an effect on the character speed of the humans in the virtual environment. To make the walking character look realistic, the character speed has to match the animation of the moving legs. When you use a character speed that looks realistic in the development environment under a high resolution, this does not have to look realistic in the published environment under a lower resolution. Because the performance is better, the characters may seem to walk faster, thus the animation of the legs does not match the speed anymore. So you have to play around with this variable until it looks right in the published environment, even though it might look a bit slow in the development environment. Aside from this part, another aspect that has a great influence on the performance is the use of high resolution textures, like we stated earlier.

Support forum

Quest3D has an extensive and well maintained support forum. We recommend future users of Quest3D to really use this forum [43]. You can lookup former questions or post your own questions. The question is usually answered within two or three days by the staff of Quest3D or some of the other users on the forum.
Importing from 3D Modeling applications

Quest3D supports the import of 3D objects from other 3D modeling applications. We used Maya (Alias Wavefront). You can install a Quest3D plug-in, which makes it possible to export from Maya to the X file format, which can be read by Quest3D. However the X file format, does not contain the textures, but references to the image-files on your harddrive. So when you want to import the X file into your Quest3D environment, you should not forget the actual image files. Otherwise Quest3D cannot read the textures. There are also plug-ins for 3DStudio Max and other 3D modeling applications.

Object oriented (groups)

Quest3D is not object oriented. There is a possibility to group a network of channels as a separate channel group and assign parameters to it. But this channel group does not have the option of a return value. Thus you cannot really construct an object. Quest3D makes use of channels that can have an input and output, but these are elementary building blocks and thus can also not be really seen as objects. This is a drawback of the application.

Array variables

We recommend to use arrays to store important variables. They work fast and easy. It is possible to store a variable in a Value channel, but when you want to update or read the variable you have to link to it or make a shortcut. When many parts of your application use this variable the links to this variable become immense and confusing. Also when you want to copy a group of connected channels, the shortcuts are not included. So you have to link them all over again. Using arrays, you can link to an Array value channel. This channel can easily be reused and copied. Also the value of the variable is easy to retrieve using the array table.

Undo function

Quest3D has no undo function. Instead it makes use of snapshots. Every few minutes automatically a snapshot is taken, saving the current state of your application. You can then return to this snapshot. You can also make manual snapshots. We don’t know the exact reason for this snapshot concept, but think that it would increase the usability if there was just a common undo function.

Hardware support

The VR edition of Quest3D is still young. They do not have support for all the VR hardware yet. As of this moment the Flock of birds (Ascension), which we are using is not supported yet. Therefore we cannot use the HMD properly to view the virtual environments.

External software support

Quest3D assumes that an external software package is used for modeling the 3D objects. Therefore it supports the import of 3D objects from modeling packages like Maya, 3DMax etc. like we saw earlier. But the use of external programming applications is not very well supported. In Quest3D you can write your own channels using a programming language, which shows a lot of similarities with C. But it is imaginable that we would like to control Quest3D with an external application in order to create for example AI agents. Quest3D has an ActiveX control that has some public methods that make it possible to access variables in Quest3D, read and write. This control also makes it possible possible to trigger an Event. You can then use Javascript, VBScript or any other ActiveX container to control Quest3D. So it is possible to use Visual basic or Borland Delphi to control Quest3D. However we have not tested this functionality yet.
Conclusion

As we see in the above, most of the comments regarding Quest3D are about the usability of the application. There are a few points that could be improved. Except for these usability points it is a very powerful application and certainly useful for this project. Another point of attention is the current insufficient hardware support. Quest3D however has announced that they are going to support the Flock of Birds in the near future. We should also take into consideration the use of an external application to control Quest3D. This way we can build the user interface externally or let Quest3D applications be controlled by AI agent software. Finally one has to consider the large performance difference between a project in the published environment and the development environment, with different resolutions.

5.5 Design

5.5.1 Modular structure

In our design we tried to have a modular structure. In contraction with our task analysis we came to the following modules:

![Modular structure of a virtual environment](image)

Figure 5.5 – Modular structure of a virtual environment

In figure 5.5 we see the modular structure of the virtual environment. The world control objects can be seen as the interface towards the user. These objects can be widgets like slidebars, buttons, radio buttons etc. The virtual world module is the actual 3D realization of an environment. Each world is constructed of a few
modules. There are two main groups, namely the world controls and the world elements. The world elements are the visual or hearable parts of the world, which are brought to the senses of the user. These world elements have parameters like position, material etc. The world controls make use of these parameters to control these world elements. So the world controls and elements communicate with each other through a two way communication channel. Each world element module has a separate dedicated world control module. These control modules handle all the input from the user and the system itself in order to supply the world elements with the proper actions. World controls do not necessarily have to have a user input. For example people that randomly walk over a square, without any input from the user, still have to have a control that calculates their AI movements (standing still, walking, going to the next destination).

Each world definitely has some world controls and world elements. But the content of these controls and elements may differ for each world.

One special world control object is the navigation, because it has no visual or auditive representation in the virtual world, but it actually controls what the participant sees. It is somewhat the control of the viewport, through which the virtual environment can be seen. This module can have user input (E.g. the therapist who is moving the patient in a certain direction). This module however doesn’t communicate with any world elements. It only uses external user input and external parameters from the tracker system. The output has effect on the virtual world.

5.5.2 Separation of the user interface and the application

The above modular structure forms the basis for a virtual environment. In the prototype world that we implemented, the user interface (world control objects) is implemented in Quest3D. In the new framework, it would be better if the user interface is separated from the rest [16]. The world elements and controls can be implemented in Quest3D and the world control objects (user interface) in a common program language. The reason for this is that it is very time consuming to build user interfaces in Quest3D. Quest3D is build to make interactive virtual environments, but it does not have extensive support for building and controlling easy to manage user interfaces. Though it is possible to work around this, we advise to use other software for the construction of the user interface. In this prototype world we build the user interface in Quest3D. This user interface will not be part of the eventual new framework, but merely serves as easy way to give input to the world controls.
6 Implementation

As we stated earlier, Quest3D is not really object oriented. An average program in Quest3D consists of a complex network of channels, ordered in directories and subdirectories. So it is quite difficult to map the modular design to the working environment of Quest3D. It is up to the programmer to bring as much structure in to his or her program as possible. In the next paragraphs we try to give an overview of the different modules and their associated channels.

6.1 Variables

Before we get into the modules, we give a short overview of the variables. Each world has some global variables. Some of the variables below will be straightforward, others will be clarified later on. For the ease of use, all these variables are stored in array tables. This way you don’t have to navigate through the complex structure of interconnected channels to find the channel that holds the appropriate variable. Another advantage of this approach is that you can link all the channels that use this variable to the appropriate column that holds the variable, without the use of shortcuts. This makes the network less complex and well-organized. The global variables are put in one row, each column holding a different variable.

Global variables

All the variables below are of the type value, thus containing a number.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Explanation</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min. Char Timer</td>
<td>This variable is used to calculate a random timer value for the character timer.</td>
<td>[0, ∞]</td>
</tr>
<tr>
<td>Range Char Timer</td>
<td>Each human character gets a random time, which counts down. When it is zero the character gets assigned a new destination. This is the range that is used to calculate a random timer value.</td>
<td>[1, ∞]</td>
</tr>
<tr>
<td>Max speed</td>
<td>Give the maximum speed that all the human characters have when they walk. This is mostly 0.05</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>#nodes</td>
<td>Number of nodes that the 3D graph uses. This variable is used for path-finding.</td>
<td>[0, ∞] int</td>
</tr>
<tr>
<td>Closest node</td>
<td>This is the node number of the node that is closest to the position of the camera (patient view).</td>
<td>[3, ∞] int</td>
</tr>
<tr>
<td>Nearby node</td>
<td>This is the number of a random node that is nearby the closest node</td>
<td>[3, ∞] int</td>
</tr>
<tr>
<td>Day</td>
<td>Tells if it is day=1 or night=0</td>
<td>[0, 1] int</td>
</tr>
<tr>
<td>Emission</td>
<td>Some 3D objects are connected to this variable and it gives the amount of self emission the object has. This variable is used to simulate day and night.</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>Weather</td>
<td>This variable sets the weather type ranging from 1 to 3 (1 = Clear / 2 = Rain / 3 = Rain &amp; Thunder)</td>
<td>[1, 3] int</td>
</tr>
<tr>
<td>Blur</td>
<td>This variable sets the amount of blur ranging from 1 to 4. 1 means minimal blur and 4 is the maximum amount of blur.</td>
<td>[1, 4] int</td>
</tr>
<tr>
<td>Stop distance</td>
<td>The distance the human character takes when arriving at a destination node. E.g. When this is set to 1 the human character will stop when it reaches the radius of 1 from the destination node.</td>
<td>[0, ∞]</td>
</tr>
<tr>
<td>Heartrate</td>
<td>This variable holds the amount (speed) of the heartrate and shortness of breath ranging from 1 to 7. Where 7 is the maximum.</td>
<td>[1, 7] int</td>
</tr>
<tr>
<td>Heartrate on</td>
<td>This is a Boolean variable to turn the heartrate on (1) and off (0)</td>
<td>[0, 1] int</td>
</tr>
</tbody>
</table>
| Vision       | This variable holds the type of vision that is currently selected.  
  (0=normal vision / 1 = blurred vision / 2 = tunnel vision) | [0, 2] int |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>People</td>
<td>This variable sets the amount of crowd on the square ranging from 0 to 7, where zero means an empty square and 7 means a crowded square.</td>
<td>[0, 7] int</td>
</tr>
<tr>
<td>Tunnel</td>
<td>This variable sets the amount of tunnel vision ranging from 1 to 4. 1 means minimal tunnel vision and 4 is the maximum amount of tunnel vision.</td>
<td>[1, 4] int</td>
</tr>
</tbody>
</table>

Table 6.1 Global variables (int: this variable holds only whole numbers)

Character variables

Besides these global variables, each human character in the world (except for the standing men, which we will see later on) has its own variables.

All the variables below are of the type value, with exception of the variable AI character position which is a vector array.

<table>
<thead>
<tr>
<th>Active</th>
<th>This variable tells if the character is active (1) and can be seen on the square or if the character is inactive (0).</th>
<th>[0,1] int</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI character position</td>
<td>This is a vector value that holds the x, y and z position of the character.</td>
<td>Vector of length 3</td>
</tr>
<tr>
<td>AI character speed</td>
<td>This variable is zero when the character is standing still. When the human character is walking this variable gradually increases till it reaches the global variable “Max Speed”.</td>
<td>[0, ∞]</td>
</tr>
<tr>
<td>Cur. Destination</td>
<td>This is the number of the node that is the current destination node of the character.</td>
<td>[0, ∞] int</td>
</tr>
<tr>
<td>Char. Timer</td>
<td>This is the initial timer value of the character.</td>
<td>[0, ∞]</td>
</tr>
<tr>
<td>Texture</td>
<td>This variable shows, which texture the character has.</td>
<td>[0, ∞] int</td>
</tr>
<tr>
<td>Countdown Char. Timer</td>
<td>This is the current timer value of the character. When it reaches zero the character is assigned another destination node.</td>
<td>[0, ∞]</td>
</tr>
</tbody>
</table>

Table 6.2 Character variables (int: this variable holds only whole numbers)

6.2 Virtual World

Our first concern for the Agoraphobia world, was how to make a 3D world of the Market in Delft. There are numerous ways to do this. We could for example measure all the elements and buildings on the market and rebuild them in the 3D world using these measurements. This method is very time consuming and almost impossible, thus we chose to use another method.

6.2.1 Method used for converting the Market in Delft to 3D

We decided not to make a 3D representation of each building. This would be too time consuming for the project and its additional value to the look of the world would be of negligible value. Instead we decided to take four planes (one for each side of the square) and put a large transparent texture on these planes. First
we took pictures of all the buildings on the square. We did this by taking a series of overlapping pictures of the front of the buildings. This resulted in a large set of pictures. These pictures also had perspective and because we wanted to use these pictures as textures for planes we had to remove all the perspective from the pictures. This was done using the crop tool in Adobe Photoshop. The pictures all had to have the same colour depth in order to get a smooth transition when we overlap them. Eventually we had to remove everything from the image that is not a part of the building. This was mostly the sky and parts of the ground. This was necessary because these parts where going to be transparent in the virtual world. Figure 6.1a shows the original picture of one of the buildings. Figure 6.1b shows the edited picture. You can see the white spaces in the lower left and right corner, which are the result of the perspective removal. Finally figure 6.1c shows the final view of this part in the virtual world.

Figure 6.1a Original picture of buildings with perspective

Figure 6.1b Edited picture of the buildings (perspective removed and colors updated)
After the buildings were constructed we had to place the elements on the square, like the statue of “Hugo de Groot”, the lampposts and more. All these objects were modeled in Maya3D and than converted to Quest3D using the Xfile-exporter plug-in from Quest3D. This plug-in is installed in Maya and allows Maya to export 3D objects and 3D environments to the X-file format, that can be imported by Quest3D. The coordinate system of Quest3D differs from that of Maya. So we had to flip the X-axis in order to get the 3D objects right. In figure 6.2a, b, c, d and e we see examples of 3D objects that were modeled in Maya 3D and then imported into Quest3D. For the church we had to make pictures of each side of the church, in order to get the right textures for the 3D model. We used an UV map to do the texturing of the church. This process was also necessary for the lampposts, the signposts and the statue of Hugo the Groot.
6.2.2 Ambience sounds

During the implementation we noticed that the ambience sounds show a strong relation with the events in the virtual environment. Therefore we decided to slightly change this aspect within the implementation. We coupled these ambience sounds to the events in the virtual environment. E.g. When the number of people on the square increases, the sound of mumbling people automatically increases.
6.3 Camera setup

6.3.1 Composition of multiple viewports

To begin with, the virtual environment is constructed of multiple camera-viewports that are mapped onto the screen. In our example we also have a separate camera for the user interface, which can be replaced by an external user interface.

![Composition of multiple viewports](image)

Figure 6.3 a – Composition of multiple (camera) viewports

In figure 6.3 a we see the setup that runs with one computer, with two monitors. The therapist view and patient view are actually identical except for their resolution. The therapist view has a smaller resolution in order to appear on the user interface. On top of these viewports there is a separate camera viewport for the simulation of thunder. This is actually a camera pointed at a white surface. This surface changes its transparency as we will see later on in the weather control and elements. The last viewport is that of the user interface. In figure 6.3 b we see an example of how a viewport camera is pointed at the user interface. All viewports are constructed this way and are combined and layered on top of each other to get the eventual composition we see in figure 6.3 a. This image is then stretched to a resolution of 1280 x 480 (2x 640x480), which is projected onto two monitors using a desktop with horizontal span.
6.3.2 Extension to HMD and TCP/IP connection

The signal or image for the patient view (the right monitor) is also the image that can eventually be sent to the HMD. This is quite easy since it is a VGA output. However this image is mono and we would like to have a stereo image when we view the world on an HMD. This extension can be made without much overhead. In Quest3D we add another camera on the same position as the patient view, but adjust its position slightly to the right. This way we get a left-eye and right-eye camera viewport. Due to the fact that Quest3D does not support our tracker (The flock of birds), we were not able to test this aspect yet.

Also TCP/IP connection is possible, where we use two computers that communicate with each other. This way we can have the user interface on one computer and the patient view on another computer. The only data that is exchanged between the computers are the variables, because the virtual environments are run locally on each computer. The variables in the agoraphobia world are all stored in arrays and can be converted to network variables. So the agoraphobia world can easily be extended to run on a network. We have done some tests to send network variables between computers, using TCP/IP and these tests were succesfull.

6.4 Crowd control and elements

This is one of the more complex modules within the agoraphobia world. The human characters have to wander and walk around on the square, sometimes stand still and randomly choose their own path.
There are two types of human characters in the agoraphobia world: Characters that walk and sometimes stand still on a random position and characters that constantly stand still on a fixed position. First we will describe the implementation of the walking characters, and after that we will explain the implementation of the characters with a fixed position.

6.4.1 Walking characters

Motion damping

The human characters are skinned characters. This means that they consist of a bone structure, which is enwrapped by the 3d object of the structure (the skin), eventually a texture is mapped onto the skin. In figure 6.4 we see an example of this skinning process. By moving these bones, the skin also moves along. Thus the character can be animated by moving these bones. A set of bone movements can be stored as a motion set, containing the animation. The walking character has two motion sets: One for walking and one for standing still. In figure 6.5, we see the channel network that handles the motions of the character. In order to use the character we need to have a smooth transition between the two motion sets. This is done by the motion blender channel. It uses the value damping channel to do the transition. When the damping value is zero the character will have the idle motion set (standing still) and when the damping value is one the character will have the walk motion set. However when the damping value is 0.6 the character will have 40% of the idle motion set and 60% of the walking motion set. So by changing this value we can control the motions of the characters. When the damping value suddenly changes from zero to one it does not immediately change. It gradually changes to the value of one, with steps of 0.2 (which is the value of the damping amount). The damping value uses the character speed variable to determine which motion set it should use.

Figure 6.4 – From L to R: 3D bone structure, 3D Skin and the eventual textured 3D result
Figure A2 in Appendix A shows the 3d graph that was used for this virtual environment. A 3d graph is a collection of nodes, with are completely or partially connected. The whole path-finding system revolves around this 3d graph. In our example the environment has a 3d graph that consists of 38 nodes in the range of [0,37]. The number of nodes is stored in the global variable #nodes. The motion planning channel gives the human character a destination node (node number) and the human character then automatically calculates a path to that node. When the character is assigned a destination node (Cur. Destination), the speed of the character is adjusted and gets the value that is stored in the Max speed variable. The motion damping reacts on this value change, by changing the characters motion from 'idle' to 'walking'. Also the orientation of the character is changed. Matrix calculations are used to make sure the character faces the destination node that it is heading for. First we take the movement vector which is a vector that is either (0,0,0) or (0,0, [maximum speed]). As you can see the character moves in the z direction. This vector is perpendicular to the front of the character (pointing into the direction the character is walking). We take the AI motion matrix, which holds the translation, rotation and scaling of the character and substract the rotation from this matrix. The result is a rotation matrix. The movement vector is then multiplied with this rotation matrix to create a matrix for the adjustment of the character heading. The result is that the character is always facing the direction it is walking in.
Character AI

The human characters have to act automatically and naturally. They have to randomly find their path through this network of nodes and sometimes even stand still. In order to reach the effect of a crowded square the characters all have to circle around the participant (camera). This is implemented in the following way.

First of all characters all have a variable called ‘Active’. When this variable is zero it means that the character is not present on the square and when this variable is one it means that the character is present and visible. This transition must be gradually. We can’t just make the characters disappear as soon as they become inactive. Nodes 0, 1 and 2 are reserved as so called home nodes. They are positioned outside of the visible virtual environment. As soon as a character is set to inactive, the character gets the assignment to move to one of the home nodes and wait there until the character becomes active again. When a character is active is depending on a timer (Countdown Char. Timer). The value of this timer is a random number that is calculated from the global variables ‘Min. Char Timer’ and ‘Range Char Timer’. In this virtual environment these variable consecutively hold the values 100 and 1000. This means that a random value for the timer is created that lies between 100 and 1100. The timer counts down to zero and each time the timer reaches zero it is assigned a new timer value and starts to count down again. Each time the timer reaches zero, the character also gets assigned a new destination node. When the destination node is reached, but the timer is not zero yet, the character will stand still at that node and waits until the timer reaches zero and the character gets assigned a new destination node to move to. If the character has not reached the destination node yet, but the timer value is zero already, the character gets assigned a new destination node and just changes direction. This way the character will move naturally and sometimes stand still. The destination node is calculated by using the two global variables: Closest node and nearby node. Each time a one of the timers of the characters reaches zero, the number of the node that is closest to the camera is calculated. The nearby node gets the number of a random node that is nearby the closest node. This is done because we don’t want to have all characters going to the same node, which is the closest to the camera. In fact, we want the characters to move to nodes that are around the camera.

A problem that arises is when two or more characters have the same destination node and all reach this node before the timer reaches zero. We then have two or more characters standing on the same spot. To deal with this problem, we defined the global variable ‘stop distance’. In our case the stop distance is one. This means that the character has successfully reached a destination node as soon as it is in a range of one
of the nodes’ position. This way characters that reach the node coming from a different direction will all
stand still on a different position. This can be seen in figure 6.6.

All the characters’ variables stored in a table with multiple arrays (rows). Each character has an own row
(index). Each character is constructed of a complex network of connected channels and subdirectories. This
whole network can be connected to an index value, making sure the network reads all the variables from
the right row. E.g. The variables of character 2 as an array of values on the second row. The network of the
character is connected to the index value two and thus the network reads and sets the character values in
row 2. This way each character has its’ own set of variables. This can easily be seen when opening the
array section in Quest3D and viewing the Character table.

6.4.2 Standing characters

The standing characters are positioned on the borders of the square. They have a fixed position and only
one motion-set. These characters can also be set visible and invisible. They are implemented using fast
skinning. This technique is mostly used to display large crowds with high performance.

Finally all these characters react on the global variable ‘people’. This variable holds a value between zero
and seven. The crowd controller automatically sets one or more character active or inactive according to the
value of this variable.
6.5 Weather control and elements

The part of the weather control and elements actually consists of two parts. One part controls the actual weather, say clear, rain or thunder and the other part controls the night and day.

Clear, Rain and Thunder

The global variable weather has three stages: 1=Clear / 2=Rain / 3= Rain&Thunder. In figure 6.7 we can see how the channel network of the weather part reacts to this variable. According to this variable the network decides for example if it has to play the sound of rain. Like we saw previously the thunder effect is created using a white square, with an alpha channel. When the value of this channel is zero this means that the white square is not visible. When the value is one the white square is completely visible. The white square covers the whole screen, because it is set up as a separate viewport, with a separate camera. The white square also has a function that decreases the alpha value to zero as soon as it becomes larger than zero. This way we only have to set the value of the alpha channel to one in order to create the thunder effect and play the sound. The thunder comes randomly. This is done by using a timer, which is given a random
value. As soon as this timer reaches zero the thunder effect takes place and the timer gets another random value. However this whole procedure is only executed when the global variable of weather is set to three.

Creating rain in the virtual environment was also a difficult task, because we wanted to have realistic rain without giving in much on the performance. At first we tried to make a separate viewport and camera for the rain. The viewport would show a transparent square covering the whole screen. Rain would then be simulated by alternating two transparent textures of raindrops at a high rate. The effect was pretty realistic as long as you did not look up. As the patients view looked upwards to the sky the drops still moved the same direction, thus seeming to go horizontal in stead of vertical. Eventually we solved this problem using particles. A particle emitter is an object that emits particles. The Particle Emitter (PE) is an advanced extension of the Particle Object channel. With the particle object you can control the shape, size, lifetime, direction and many other values of the emitter. Particle systems are mostly used in 3D environments to create nature elements like fire, smoke or explosions. We used a simple texture of a white vertical line to simulate the raindrops. The emitter was then put outside of the virtual environment, emitting particles in the air that came down in the visible part of the virtual environment, thus creating the effect of rain. This technique seemed to work fine, without any real decrease in system performance.

For the effect of a wet ground we created an emissive channel. We will discuss that in the next section about day and night.

**Day and Night**

This is the other part of the weather module. It simply depends on the global variable Day, which is zero by night and one by day. According to this value the texture of the sky-sphere is changed. However when it rains, we didn’t want to have a clear sky, so it also reacts on the value of the global variable weather, thus giving us a grey sky as soon as it rains or thunders.

In figure A3 from appendix A, we see an example of a 3D object. All the 3D objects in Quest3D have globally the same structure. The value motion channel represents a 4x4 matrix containing the position, rotation and scale. The surface channel contains information regarding the texture, the geometry and the material of the object. The material channel amongst others contains the information regarding the diffuse, emissive and specular values of the object. Diffuse is the color that is calculated using the lights diffuse color in the current scene. Emissive color can be used to make a rendered object appear to be self-luminous. Specular color is used to give a shine to the object that works the same way as diffuse in that it is dependent of lights in the scene.

In reality objects receive less light by night, thus reflecting less light, making them look dark to the human eye. We wanted to create this same effect in the virtual environment. We implemented this making use of the emissive channel. For some objects the value of the emissive channel is calculated by making use of the global variable emissive. Thus the buildings for example will look dark in the evening and light in by day. The same technique is used to give the ground a wet look when it is raining. This is also automatically calculated using the global variables Day and Weather.

**6.6 Vision control and elements**

The vision control and elements consist of two main groups: The tunnel vision and the blur vision.

**Tunnel vision**

The tunnel vision partially uses the same technique as the thunder. We set up a separate camera pointing at a partially transparent surface. This surface spreads across the whole screen and overlaps the complete
virtual environment. This surface has one texture that is completely black and four different alpha textures. In figure 6.8 we see an overview of these four textures.

The black part of the alpha textures is completely transparent and the white part defines the part of the image that is completely visible. The gray parts show gradual transparency. Because the main texture is black, these alpha textures show the world in the middle black circle, turning gradually black towards the edges. According to the value of the global variable tunnel, one of the four alpha textures will be used.

This is the first part of the tunnel vision. The second part deforms the virtual world, using a matrix operation on the walkthrough camera. We used a DirectX projection matrix, with three variables: Near clipping plane distance, far clipping distance and zoom factor. According to the value of the global variable tunnel, these parameters are given a different value, resulting in a deformation of the patient view. In figure 6.9 and figure 6.10 we respectively see the channel network for the deformation and the associated projection matrix.

![Figure 6.8 – Tunnel vision gradations (Alpha textures)](image)

![Figure 6.9 – Projection matrix parameters (Setting values)](image)
Figure 6.10 – Projection matrix for the camera deformation

### Blurred vision

In figure A6 in Appendix A, we see the channel network for the Blur vision. It consists of four blur groups numbered 1 to 4. According to the value of the global variable ‘Blur’ a connection is made to one of these blurs. The blur effect is implemented, using an iterative process of down sampling, with multiple rendering layers on top of each other. The Blur effect, has an effect on all the previous layers in the z-buffer, so in our rendering of the virtual environment, we had to put the blur rendering after the rendering of the world, but before the rendering of the user-interface. If this is not done, the user interface would be blurred as well. In figure 6.11 we can see the gradations of the blurred vision.

Figure 6.11 – Blurred vision gradations

### 6.7 Heartrate & Breathing control and elements

According to Emmelkamp [7] the simulation of an increasing heartrate and breathing, has an anxiety provoking effect on the patient. This is mainly because the patient recognizes these bodily sensations and thus evokes the anxiety. There was some discussion if the heartrate and breathing should increase and decrease simultaneously. In the interview we had with Emmelkamp, we came to the conclusion that it would be realistic if the heartrate and breathing increase and decrease simultaneously. In the real world they are also closely connected.
The simulation of the heart rate and breathing is done using multiple sets of sounds. Which of these sets to play, depends on the value of the global variable heart rate. In figure 6.12 we see a screenshot of the associated channel network. We see how the heart rate sounds and breathing sounds are setup separately but react simultaneously on the change of the global variable heart rate.

### 6.8 Navigation

Everything the user sees is viewed through a Walkthrough camera. This is a camera that has 6 DoF, in order to navigate the world. This camera has to be attached to a collision object, to represent the ground. If this is not done, the camera would make an endless fall. The camera also has a collision sphere. The radius of this sphere defines the height of the viewer, but also the distance to which the camera will react on a collision. Besides the collision object of the ground, we also implemented a collision object in the form of a rectangle, to make sure the user cannot just simply walk through the houses.

In the current system, navigation is done using a fixed path. The therapist can navigate the patient forward and backward on this path using different speeds. We experimented with this type of navigation using Quest3D. We implemented an example of a staircase. The camera has a fixed path, that runs from the ground, up the stairs to the top of the staircase. In Appendix A, figure A7 we see a screenshot of the staircase, with the camera. In our example the speed and direction can be adjusted. In Quest3D this is quite easy to implement. For the virtual world of the square in Delft, we choose to use another navigation. Using a fixed path on a square was not suitable, because there are too many directions you can go in. Instead we
choose to use a free navigation, where the therapist can move in all of the four directions and the patient can look around using the HMD.

At the time of our implementation, Quest3D still had no support for the tracker (Flock of Birds). Therefore we could not use the HMD to view the world. We solved this problem, using the mouse as an input for the rotation instead of the tracker. In Quest3D the input values of a mouse, a joystick and a tracker are the same. So at the time the Flock of Birds is supported by Quest3D, the mouse input can easily be replaced by the input of the tracker.

6.9 User interface (World Control objects)

We previously stated, that it would be better to implement the user interface outside of Quest3D. One of the reasons for this, is the overhead it takes to make a user interface in Quest3D. Quest3D is a well suited program to make interactive virtual environments, but they don’t have many tools to implement a graphical user interface. In figure A8 in Appendix A, we see an example of the user-interface. All the elements (widgets) on the user interface have to be manually modeled, using 3D objects. Like we saw earlier we used a separate camera which is projecting the user interface. In front of the camera is a surface that holds the texture of the background of the user interface. The widget of the slide-bar, for example, consists of two 3D objects: The bar itself and the slider. In order to detect a mouse-down on the slider, a lot of calculations have to be made. First the 2D position of the mouse pointer has to be calculated. Then the 3d position of the slider has to be converted to a 2d screen position, using a matrix calculation. Also the range of the slider has to be programmed using a large network of different channels. We wanted the slider to move along with the mouse pointer, as long as the mouse was down. Also there had to be some value changes in the global variable array as a reaction to the position change of the slider. In figure A5 in Appendix A, we see the channel network for the user interface. Even though the user interface is not very complex, the network is. The channel network that is shown, does not even show the multiple subfolders. One can imagine the overhead it would take to construct a complex user interface. For the completeness we added a screenshot of the user-interface in Appendix A, figure A8.

6.10 Evaluation

After the completion of the implementation we evaluated the virtual environment with a therapist\(^1\) of a clinic of psycho-medical care in the Netherlands. Together with prof. Emmelkamp the therapist was involved in previous projects at the Technical University in Delft regarding VRET. The therapist tested and evaluated the worlds for “fear of flying”, claustrophobia and more. We choose to evaluate this world with therapists instead of patients, because the therapists are specialists on the phobia and are also end users of the application. The therapist has experience with the current framework and current worlds, but the new agoraphobia world which we evaluated, was completely new to her. We had a set of questions regarding the user interface and the virtual environment itself. It must be said that the user interface that was used for this prototype is not necessarily the same user interface that is going to be used for the new framework. The comments of the therapist can however be used for the design of the new user interface. We asked the therapist to point out what she thought of the different elements that we showed here. We first demonstrated the element and subsequently recorded her comments and recommendations. Next we will give an overview of the comments by the therapist, grouped by subject.

\(^1\) J. Wiersma (PsyQ clinic of psycho-medical care in the Netherlands)
We asked her opinion about the controls as well as the world elements of the weather. As far as the weather controls and elements, the therapist had no comment. According to her the current state of the Rain, Thunder, Day and Night elements is fine.

Crowd controls and elements

The therapist stated that it would be a surplus value if there were more characters on the square. Some of the agoraphobia patients are afraid of an empty square with a few people, others are only afraid when the square is really crowded. So the virtual environment would be anxiety provoking for the first group, but less anxiety provoking for the second group. She came up with the idea to create a concentrated, group of people somewhere on the square. The therapist can then navigate the patient to this heavily crowded spot. This would also provoke anxiety for the second group, thus solving the problem of too few people on the square. It would also save performance costs of the system, because we only crowd a small piece of the square instead of the whole square.

She also recommended to combine this virtual environment, with another agoraphobia specific environment: The queue. It would be quite easy to implement a queue of characters standing in line in front of one of the shops. The therapist can then take the patient to the queue to stand in line.

Heartrate&Breathing controls and elements

The therapist thinks the volume level of the heartrate is a little too high in proportion with the volume level of the breathing. She recommended decreasing the volume level of the heartrate a bit. We also asked her opinion about the different levels of the heartrate&breathing speed. She stated that one higher level could be added, to the range of the heartrate&breathing.

Vision controls and elements

According to the therapist, the minimal Blur was the most realistic. She recommended shifting the minimum and maximum blur to the left, making the maximum value less blurred.

As far as the control elements the therapist stated that it would be nice if the Blurred vision and tunnel vision could be used simultaneously. In the current user interface the three types of vision (normal, blurred and tunnel) can only be used one at a time.

Navigation

In the current navigation the patient moves in the viewing direction. She recommended to separate the navigation from the viewing direction. This way the patient can rotate his or her head, to view the virtual environment, without having any effect on the direction the therapist wants the patient to go.
For the completeness we summed the above evaluation in a table. (++ "outstanding", + "sufficient", ++ "ok, but could use some adjustments", - "needs adjustment", -- "insufficient")

<table>
<thead>
<tr>
<th>Module</th>
<th>--</th>
<th>-</th>
<th>++</th>
<th>+</th>
<th>++</th>
</tr>
</thead>
<tbody>
<tr>
<td>World controls</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vision control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heartrate &amp; Breathing control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weather control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crowd control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Navigation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>World elements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vision elements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heartrate &amp; Breathing elements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weather elements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crowd elements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.5 Result of the evaluation by therapist

**Conclusion and Further recommendations**

The suggested features were not implemented in the world. The purpose of this evaluation was to get a better understanding of what the virtual environment for the treatment of Agoraphobia needs. By looking at the results of this evaluation we see that we have to pay more attention to the navigation, the control of the heartrate & breathing, but especially to the crowd elements, because that is an important fact of the agoraphobia world. As a whole the therapist had a positive feeling about the effectiveness and usefulness of the virtual environment. We also asked if she had any further recommendations. She finally stated that the option of an ambulance sound, would also contribute to the virtual environment, because this sound provokes a lot of anxiety with agoraphobia patients.
7 Conclusions and recommendations

7.1 Conclusions

During this project we designed and evaluated anxiety-provoking virtual environments for the treatment of agoraphobia based on requirements from therapists. We implement a “prototype virtual environment” for the treatment of agoraphobia, based on these requirements, which can serve as an impulse towards a new framework for VRET of phobias. Finally we gave an evaluation of the software and techniques that were used to implement the “prototype virtual environment” for the treatment of agoraphobia.

We have done interviews and evaluations with therapists. This resulted in a list of requirements and recommendations. We managed to implement most of the requirements, in order for the virtual environment to serve as a prototype for virtual environments that will be build within the future framework. The level of habituation in this prototype virtual environment could be improved. In the current virtual environment the level of habituation can be decreased by the therapist changing the look of the world, using the world controls, but it would certainly be a surplus value if multiple versions of this environment were created or if the size of the environments could be adjusted. This way the patient does not have the chance to get used to one virtual environment. Another requirement Emmelkamp gave, was that the therapist should be able to change the ambience sounds. During the implementation we noticed that the ambience sounds show a strong relation with the events in the virtual environment. Therefore we decided to slightly change this aspect within the implementation. We coupled these ambience sounds to the events in the virtual environment. E.g. When the number of people on the square increases, the sound of mumbling people automatically increases. Furthermore of all the requirements there is a minimum of requirements that should be met in order to have an anxiety provoking virtual world:

Common requirements
- Sufficient resolution 640x480, sufficient FoV (70 degrees)
- Two Sensorial modalities (Visual- and auditory sensorial modalities)
- Existence of other creatures in the environment.
- High level of interaction (6DoF)

Phobia specific requirements
- Simulation of a Phobic situation (in this case a square which is a feared situation for agoraphobics)
- Heavily adaptive virtual environment (for less habituation)
- Simulation of bodily sensations

In this thesis we also evaluated the software and techniques used to implement this virtual environment. Quest3D, which was used to implement the virtual environment, has many advantages, but also some disadvantages. If we are going to use Quest3D for the implementation of virtual environments in the future framework, we have to take into account the complexity that it brings. A simple virtual environment, with a few functionalities can quickly become a complex network of interconnected channels in folders and subfolders. Also the fact that Quest3D is not object oriented is a main drawback, for using Quest3D in the future framework. The purpose of a framework is to provide a logical structure for classifying and organizing complex information. Because Quest3D is not object oriented, the information will remain somewhat complex. However you can build around this by using a model during the design of the system, which we showed in paragraph 5.5.1. Finally the implementation of a user interface with Quest3D is not recommended. In paragraph 6.9 we showed how digressive it is to build a user interface, using Quest3D.

For the virtual environment to be used in the future framework a few adjustments have to be made: The function of the current user interface was merely to test the functionality of the world and the software. The current user interface is not going to be the eventual user interface for the future framework. Therefore a new interface has to be built.
Furthermore we were not able to test the world on a HMD, because of limited hardware support by Quest3D. This should also be taken into account. Quest3D has announced that they are going to support the Flock of Birds in the near future, but until the support is actually there this causes an uncertainty during the design and implementation of the future framework.

### 7.2 Future recommendations

During our interviews and evaluation we collected a few future recommendations, which will be discussed in the following. For the agoraphobia world it would be a surplus value if the sounds or visuals of an ambulance and a hearse were implemented. According to Emmelkamp this provokes a high level of anxiety for agoraphobia patients. It would also be a surplus value if a concentrated group of people was created somewhere on the square. The therapist can then navigate the patient to this heavily crowded spot. This would also save performance costs of the system, because we only crowd a small piece of the square instead of the whole square. The current system consists of a collection of different files on each computer. In order to make the system work these files have to be placed manually in directories and subdirectories on different hard drives. It would be an improvement if the files could reside in one directory or in the optimal case could automatically be placed in a proper destination using an installer. Finally, it would increase the usability of the system, if the user (therapist) could switch between environments, without first having to shut down the application and run a new one. In the optimal case the different virtual environments are interconnected. For example: One can move from a square to take the subway, that brings you to the airport, where a flight can be taken.
8 References


16. Mast, C. van der, Versendaal, J., “Separation of user interface and application with the Delft Direct Manipulation Manager (D2M2)”, IFAC Man-Machine Symposium (1992), pp. 3.1.2.1-3.2.6
17. Mast, C. van der, Elaborated case for MKE project 1, IN1810, Delft University of Technology, Faculty EWI, MMI, November 2004.


34. Website: http://www.ncsa.uiuc.edu/VR/VR/cave_software.html
35. Website: http://www.hitl.washington.edu/scivw/EVE/I.C.ForceTactile.html

36. Website: http://www.hitl.washington.edu/

37. Website: http://www.vrphobia.com/

38. Website: http://www.previsl.com

39. Website: http://www.rustybrick.com/definitions.php

40. Website: http://www.hitl.washington.edu/scivw/EVE/IV.Definitions.html

41. Website: http://www.science.org.au/nova/021/021glo.htm

42. Website: http://www.uhnres.utoronto.ca/ehealth/html/glossary/eh_glossary.shtml

43. Website: http://www.quest3d.com/index.php?id=30

44. Website: http://mmi.tudelft.nl/~vrphobia/

9 Abbreviations

AR – Augmented Reality
CAVE - Computer Automatic Virtual Environment
DoF – Degrees of Freedom
GUI – Graphical User Interface
HMD – Head Mounted Display
PDA – Panic Disorder and Agoraphobia
RW – Real World
SUDS - Subjective Unit of Discomfort Scale
VE – Virtual Environment
VED - Virtual Environment Desensitization
VR – Virtual Reality
VRET – Virtual Reality Exposure Theraphy
VW – Virtual World
Appendix A – Quest3D Screenshots

Figure A1 – Quest3D Screenshot of the channel network of a human character in Quest3D
Figure A2 – Quest3D screenshot of a 3D Graph

Figure A3 – Quest3D screenshot of a 3D object network
Figure A4 – Overview of the application (excluding subfolders)

Figure A5 – User interface (excluding subfolders)
Figure A6 – Blur vision

Figure A7 – Screenshot of the Staircase world (development environment)
Figure A8 – Screenshot of the therapist user interface