TITLE: A Virtual Umbilical Cord

AUTHOR: Ausiàs Pomés Freixa
RESEARCH DIRECTOR: Mel Slater
INSTITUTION: EVENT Lab, Universitat de Barcelona, Departament de Personalitat, Avaluació i Tractaments Psicològics
ACADEMIC ADVISOR: Carlos Antonio Andújar Gran
DEPARTMENT: Dept. Llenguatges i Sistemes Informàtics

DEGREE: Enginyeria Informàtica
FACULTY: Facultat d’Informàtica de Barcelona (FIB)
UNIVERSITY: Universitat Politècnica de Catalunya (UPC)

DATE: June 26, 2011
Contents
1. Abstract ................................................................................................................................. 7
2. Background ............................................................................................................................ 8
   2.1. Body representation and body perception ................................................................. 8
   2.2. The rubber hand illusion ............................................................................................ 9
   2.3. The virtual hand illusion ............................................................................................ 9
   2.4. Presence in virtual reality .......................................................................................... 9
   2.5. Out of body illusions ................................................................................................. 10
3. Hypothesis ............................................................................................................................ 13
4. Virtual reality system to study human body perception ...................................................... 15
   4.1. Requirements analysis ............................................................................................... 15
      4.1.1. Functional Requirements .................................................................................. 15
      4.1.2. Non-functional Requirements ............................................................................ 16
   4.2. Software Requirements Specification ...................................................................... 18
   4.3. Software Design ......................................................................................................... 20
      4.3.1. The Avatars ........................................................................................................ 23
         4.3.1.1. Upper Body Part IK..................................................................................... 24
         4.3.1.2. Arms IK ....................................................................................................... 26
      4.3.2. The Cord ............................................................................................................. 26
      4.3.3. The Tweener ....................................................................................................... 27
      4.3.4. Tapping .............................................................................................................. 29
         4.3.4.1. Manual tapping .......................................................................................... 30
         4.3.4.2. Automated tapping ..................................................................................... 31
      4.3.5. Shaders .............................................................................................................. 33
         4.3.5.1. Shadows .................................................................................................... 34
4.3.5.2. Geometric Distortions ................................................................. 34
4.3.5.3. Normal Mapping ........................................................................ 35
4.3.5.4. Reflections ................................................................................ 35
4.3.6. User Interface ................................................................................ 36
4.4. Software Implementation ............................................................... 37
  4.4.1. The Cord ..................................................................................... 37
  4.4.2. The Avatars ................................................................................ 39
    4.4.2.1. Upper body Part IK ............................................................... 39
    4.4.2.2. Arms IK .............................................................................. 40
  4.4.3. The Tweener .............................................................................. 40
  4.4.4. Tapping ...................................................................................... 43
    4.4.4.1. Manual tapping .................................................................. 44
    4.4.4.2. Automatic tapping .............................................................. 45
  4.4.5. Shaders ....................................................................................... 47
    4.4.5.1. Shadows ............................................................................ 48
    4.4.5.2. Geometric Distortions ....................................................... 49
    4.4.5.3. Normal Mapping ............................................................... 49
    4.4.5.4. Reflections ....................................................................... 50
  4.4.6. User Interface ............................................................................ 51
4.5. 3D scene design .............................................................................. 51
  4.5.1. The avatars .............................................................................. 51
    4.5.1.1. Modeling .......................................................................... 51
    4.5.1.2. Texuring .......................................................................... 51
    4.5.1.3. Animation ....................................................................... 52
  4.5.2. The Cord ..................................................................................... 53
1. Abstract

The present document explains the work that has been done to create a virtual reality software and hardware system with the aim of carrying out a scientific study in the field of body representation.

This project involves aspects of computer science, neuroscience and psychology. We dedicate the first pages to briefly present the background and previous work for each of those fields in relation to body representation.

We will especially focus on the computer science aspects. The document will focus in detail on the design and implementation of the software that has been developed and its integration with the different hardware systems needed to prepare the virtual reality system for experimentation on human subjects. This document also describes in detail all the work related specifically to computer graphics. First, it details issues of modeling, texturing and animation. It also explains how shaders are used to provide the solution to some requirements of the experimental studies. Shaders create a set of special effects to provide visual feedback to subjects and to increase the realism of the 3D environment. A shader is a program written in a C-like language specific for shaders that is compiled independently from the main application and executed on a GPU¹ (Graphics Processing Unit). Shaders allow for customization of the graphics pipeline and have important applications in computer graphics and general purpose computing tasks (GPGPU, General-Purpose computation on Graphics Processing Units).

The last part of this document explains the design and methodology we used to carry out initial pilot studies and to finally carry out the actual experiment. It also describes the different ways we used to gather experimental information through the use of questionnaires and physiological measures.

¹ GPUs are graphics processors present in most of the modern graphic boards for personal computers, mobile devices, game consoles and others. Their main task is to accelerate 2D and 3D graphics in computers and computer-like devices.
2. Background

2.1. Body representation and body perception

As humans, we usually know the physical limits of our body. Our brain has multiple representations of our physical body, for example the homunculus [1]. This representation is usually called body schema [2]. This body schema is the person’s mental or internal representation of his or her body. The body schema holds several properties such as the perceived body shape, size and relative position of the limbs, which is called proprioception.

We also know that the body schema is not fixed and permanent but subject to change [3]. It has some plasticity and it may be changed by external factors. Distortions in body schema may be the result of some exposure to media such as external images and video, or fully immersive technologies such as virtual reality (VR). In our study, we use VR techniques to take advantage of the brain’s plasticity in order to examine particular alterations of the body schema.

![Figure 1. Two frames extracted from the movie “Hulk” (2003). Nick Nolte, Bruce’s (Eric Bana) father, is extending the limits of his body by assimilating the essence of the matter of the objects he touches.](image)

According to [2], people immersed in virtual environments with some degree of embodiment in a virtually presented body may feel the interaction of three bodies: the objective body (physical body), the virtual body and the body schema (phenomenal body). The objective body refers to the physical body, which is defined by its measurable and objective properties with a specific geometry and topology. The body schema is the representation our brain has of our physical body. It refers to the size and shapes we perceive as well as the relative position of our limbs among others. The virtual body refers to that virtual representation we see inside a virtual environment that is replacing your real body—for instance, an avatar—. In a VR experience, the objective or physical body will compete with the virtual body [2] to influence the phenomenal body.
2.2. The rubber hand illusion

Different illusions can be elicited in an individual having an experience that tricks the brain into thinking that the body schema is different than the usual one. A very famous example that illustrates these kinds of illusions is the rubber hand illusion. It used to be a joke in some parties, but in 1998 M. Botvinick and J. Cohen published the article “Rubber hands “feel” touch that eyes see” [4] . The publication details an experiment in which a person is tricked into believing that a rubber arm actually belongs to his or her body. The setup consists of a rubber arm collocated at a distance and position anatomically correct in relation to participants. The participants’ real arm is hidden from their sight. By synchronously tapping on the real and the rubber hand for a few minutes most subjects have the illusion that they are feeling the touch from the location of the rubber arm rather than their real one, and that the rubber arm feels as if it is their arm.

2.3. The virtual hand illusion

Following the studies on body perception and looking for a similar kind of illusion, Slater et al. [5] found in 2008 that a similar illusion could be elicited on participants by means of virtual reality techniques. Their study found that participants that were being tapped synchronously on their hidden real hand and seeing a ball tap on a virtual collocated hand had a significantly higher feeling of ownership towards the virtual arm than the ones experiencing a similar but asynchronous condition. That was one important step out of many to come in the following years that would highlight virtual reality as a key technology to study body perception and ownership.

2.4. Presence in virtual reality

Although people know that they are not physically in a given VR scene and virtual events happening in that scene are not happening in the physical world, they tend to react as if that experience was actually occurring. Most of the studies that take place in a virtual environment would not be possible without this type of response—a concept called the sense of presence—.

As defined in [6] , presence is “the strong illusion of being in a place in spite of the sure knowledge that you are not there”. In other words, it is a measure of the extent to which people react realistically within a virtual environment.

The definition of presence can lead to confusion, since it is a word that has been used in other fields to refer to different concepts. For this reason, rather than referring to presence, we
prefer to use two key concepts that were introduced in [6]. These two concepts are orthogonal and they can quantify the extent to which people may experience a virtual experience as something real.

The first concept is called Place Illusion (PI). It is defined as the strong illusion of being in the virtual environment in spite of knowing that it is not real but generated artificially. That is the concept that would match the definition of presence described above.

The second concept is called Plausibility Illusion (Psi). This concept refers to the illusion of perceiving virtual events as real in spite of knowing beforehand that a computer is artificially generating those events. Psi describes how likely is that events happening in the virtual scene are actually occurring—potentially having real consequences as they would in our physical world—.

The stronger PI and Psi are in the virtual experience, the more realistic will people’s response be in the virtual environment and the more immersive will the experience be.

In general, we can say that Immersive virtual reality (IVR) seeks to enhance those two factors in order to get people to react at some level in a realistic way during the virtual experience. In our study, as in many other studies based on VR technology, it is important to induce strong PI and Psi in people taking part in the study. Therefore, from the early stages, we try to enhance PI and Psi with every single detail in the design of the experiment to create an experience with these factors as strong as possible. That will definitely constitute an essential part of our body perception study.

2.5. Out of body illusions

We have already discussed briefly some body perception issues in 2.1. The present study pretty much focuses on the way our brain perceives our limbs, our body, and the physical structure we inhabit.

Our brain normally provides us with the conscious experience of living within our bodily borders. We know very well the limits of our body, which parts of the environment belong to our body and which ones do not. We have the feeling of being within our body. We call this feature *embodiment*.

Nevertheless there are people who have reported having unusual perceptual changes in this relationship between the body and the self. They report having changes in the visual
Background

perspective and the feeling of being beyond the physical limits of their body. Moreover, in some cases people reported the feeling of being located outside their bodily limits and even seeing their own body from a third person perspective. They experienced an Out-of-body experience (OBE) [7].

There are several types of OBEs. The main characteristics that define these experiences are the sense of disembodiment (feeling outside of one’s own body) of the self to an extracorporeal location, a change in the visuospatial perspective that moves outside the bodily limits and the sight of one’s own body from that new point of view.

In 2007, Lenggenhager et al. [8] showed that the spatial unity between the self and the body can be disrupted by providing participants with conflicting visual-somatosensory input. In fact, they were able to induce some features characteristic from OBEs like “illusory self-localization to a position outside one’s body [...]”. Their study showed that bodily self can be dissociated from one’s physical body position. In their experiment, participants were wearing a head-mounted display (HMD). In one of the conditions they could see through the HMD their own body from behind. In other words, they could see their own back in front of them. The conflicting visual-somatosensory input consisted of tapping applied on their back while they were seeing the tapping synchronously applied on the back of the virtual body (their own body projected) standing in front. The following figure illustrates that setup.

![Figure 2](image.png)

Figure 2. The image shows the setup in one of the conditions were participants could see their virtual own body standing in front of them. Extracted from [8]

Their own real body was being actually recorded from their back and projected through the HMD. Two more conditions were also tested. In one of them participants could see in front of them a fake virtual body instead of their own body. In the other condition, participants had
Background

just a virtual object in front. In every condition participants had synchronous tapping and asynchronous tapping as the control condition.

Also in 2007, a similar study was carried out by H. Henrik Ehrsson [9]. Ehrsson tried to induce OBEs in the laboratory by means of video techniques pretty much in the same way as Lenggenhager. Ehrsson also recorded the back of participants, who were sitting on a chair this time, and projected those images in real time through a couple of HMDs to give participants stereoscopic vision.

![Figure 3. Ehrsson tapping the participant’s chest. Extracted from [9]](image)

Synchronous and asynchronous tapping was applied to the chest of the participants and their virtual own bodies. In the synchronous condition participants reported the experience of sitting behind their own physical body, looking at their own body from a position behind it.

On that study, computer generated experiences through the use of virtual reality showed important advantages like being able to move one’s point of view outside one’s physical body position to induce the illusion of being outside one’s body by using multisensory correlations.
3. Hypothesis

Slater et al. had several attempts at reproducing the results from Lenggenhager et al. in the study from 2007 [8] without success. That is the main motivation for our current study. In our study we want to test again the scenario that was used in the Lengenhagger et al. experiment. This time we use a computer generated 3D virtual environment as opposed to a non-stereoscopic video camera.

Our hypothesis is that once again we are not going to be able to reproduce the results in [8]. We think that plausibility (Psi) plays an important role in the experience and while the brain seems to be able to attribute ownership to fake body limbs placed in the peripersonal body space\(^2\) [8] or even a whole body seen from a first person perspective, it may not be able to attribute the same ownership to disjoint body parts or bodies outside the peripersonal space defined by one’s own body.

In [9] we do not see the same Psi issues since the tapping seen and felt by subjects remains in a plausible position with respect to the visual ego-centre. That is not the case in the setup tested in [8].

In the study presented here, we also go beyond that classic scenario by adding a second condition. In this condition subjects also have a virtual collocated body (Av1) in addition to the virtual body in front (Av2), and both bodies are connected by a virtual connection that resembles a big umbilical cord.

---

\(^2\) The peripersonal space is defined as the space immediately surrounding our bodies.
This second condition also has a tapping stage where the tapping starts first on the chest of participants instead of starting on the back of the body in front like in the classic condition. While there is tapping on the chest moving slowly down to the stomach area, participants see that the virtual tapping is moving from the chest of the collocated virtual body to the cord and along it. Eventually the virtual tapping reaches the end of the cord and moves to the back of the virtual body in front. At that moment, participants feel that the tapping moves from their belly to their own back. So, they end feeling the tapping on their own back while seeing the tapping in the virtual scene happening on the back of the body in front, just like in the classic condition.

In the condition with the cord, we hypothesize that this virtual connection may help to attribute ownership over the virtual body in front than in the condition with no cord (and no collocated body). We also expect participants to attribute some ownership over part of the cord when there is tapping on it in the synchronous condition.
4. Virtual reality system to study human body perception

4.1. Requirements analysis

Before starting any development it is always necessary to analyze the requirements our project will need to meet. We can classify all our requirements into 3 main groups. A first group of requirements comes from the fact that we are interested in investigating whether the results in [8] can be reproduced. A second group comes when defining the paradigm where we use the virtual cord. Finally, the third group of requirements corresponds to the pure technical aspect of the project, where hardware limitations and complexity of the setup play an important role in defining some specific features.

In this research project, the system was constructed through several iterations until it was ready to be used in experiments. Thus, from the outset we gathered initial requirements that were subject to change over time.

4.1.1. Functional Requirements

- The system shall keep track of subjects’ head translations and rotations. This will allow the VR system to apply the changes in position and rotation from the head movement onto the virtual cameras used as the input for both eyes in the HMD. When subjects will move their head to look to their right, they will be seeing the virtual objects placed on their right in the virtual world, just as they would expect to happen in the real physical world. That is an essential feature that will provide our VR system with visuo-motor contingencies, increasing dramatically the PI.
- The system shall move the upper part of avatar’s body according to the movement of subjects’ upper body part.
- The system shall keep track of the hands of the subjects.
- The system shall move the arms of avatars in the virtual world according to subjects’ real arms movement.
- The system shall feature real time modification of the virtual cord geometry. The virtual cord geometry has to be easy to change in real time so that we can try the experiment in different conditions like cord length and cord thickness.
- The system shall get data from a breathing device attached to a physiological device and provide visual breathing feedback to the subject.
The system shall provide visual feedback about subjects breathing in two ways. The cord shall change its thickness according to the participant's breathing. The avatars' chest shall be animated according to the breathing cycle by increasing or decreasing the volume enclosed by the mesh around the chest.

The system shall perform virtual tapping on avatars either manually or automatically. The system shall endow the virtual tapping with physicality through the use of haptics. Subjects will be feeling the virtual tapping in their body through some kind of tactile stimuli.

The system shall perform tapping in two modes: synchronous and asynchronous. In the synchronous mode the visual virtual tapping events will match the haptic feedback, with subjects feeling tactile stimuli in their body whenever they see the tapping object to touch the avatar's body. In the asynchronous mode the virtual tapping will not match the tactile stimuli breaking the visuo-tactile correlations.

The system shall play a threatening action at the very last stage of the experiment. This threatening action will allow us to study subjects' reactions by using questionnaires and physiological data.

The system shall record head tracking and electrocardiogram (ECG) data for post processing and analysis once experiments are finished.

The system shall implement the two body perception paradigms we want to study: the paradigm from 2007 and the virtual cord paradigm.

The system shall implement a minimal user interface to allow us to trigger the different actions the system can perform.

The system shall switch from one experimental condition to another by stroking a key.

4.1.2. Non-functional Requirements

General requirements:

- The scenario displayed by the system shall include as few distractions as possible in order to not introduce unknown factors during experiments that could affect our data. We shall avoid populating the scene with lots of objects and decorations when they do not have a specific purpose.

- The system shall provide subjects with a realistic experience up to a certain degree. We do not intend to create a fully realistic experience, which otherwise would not be feasible. We rather look for a plausible experience that enhances the Psi.
Virtual reality system to study human body perception

- Subjects shall be standing in the physical world. Walking is not necessary in either of the two paradigms we intend to study.

Software requirements:

- The system shall display stereoscopic images through the HMD. There will also be a mono mode with one single OpenGL viewport to develop the project though.
- The system must work in real time. The system will have to deliver at least 60 frames per second (FPS) in order to run smoothly on the HMD.
- The system shall be built using the framework XVR (eXtreme Virtual Reality; this framework is described in 4.3) from VRMedia [10]. Thus most of the work will be implemented using the XVR scripting language. This is to maintain compatibility with the work of the rest of the laboratory.
- The system shall enhance body ownership feelings whenever it is possible through the use of visuo-motor and visuo-tactile contingencies.
- The system shall use shaders to implement advanced graphic features. Shaders shall be coded using the OpenGL Shading Language (GLSL) [11]. Shaders are used in order to cheaply provide certain features to run in real time that would otherwise be impossible.

Hardware requirements:

- The system shall be working in an HMD nVisor SX111 from Nvis Inc. [12].
- The system shall integrate two tracking systems. The InterSense IS-900 [13] is an inertial and ultrasonic tracking system that will be used to track head movements. The Optitrack is an optical infrared tracking system with 12 FLEX:V100R2 cameras [14] that will be used to track subjects’ hand movement.

Figure 6. InterSense IS-900 tracking system. Image extracted from intersense.com
4.2. Software Requirements Specification

Our product will have two users. The first user is the system itself, which will trigger some functions automatically in order to have an automated sequence to run during the experiment and reduce the number of errors in run time. The other user is the system operator, or researcher carrying out the experiment. During the experiment, this person will need to do a quick setup just before running the experiment to select the proper condition or perform an initial calibration. We refer to this person subsequently as the operator.

The following Use Case diagram shows the main use cases our system needs to have in order to meet the requirements.

![Use Case Diagram](image)

**Figure 7. Use cases for user and system actors**

The behavior for each use case is described in the following paragraphs:

**Calibrate scene parameters:** only the operator can perform this calibration. At the beginning of VR experiments it is usually necessary to run a quick calibration to get some initial
parameters correct in relation to the subject physical configuration. In our case, we have the following use cases:

- Calibrate head position: it adjusts the tracked position of the subject’s head to match the position on the virtual camera in the VR scene.
- Calibrate hands position: it initializes the virtual arms position to match the position of subjects’ real arms.
- Adjust world orientation: it computes the world rotation around the Y axis (in an OpenGL coordinate system) so that subjects are facing to the right direction in the VR scene.

**Change cord appearance:** either the operator or the system can perform several actions related to the cord. For instance, both actors shall be able to change in real time the dimensions of the cord. The system needs to be able to do this in order to provide visual breathing feedback to subjects. The operator uses it to interactively set different sizes to the cord and use them as different conditions. At some point we may want to try different cord configurations to see if one works better with subjects than others.

- Show/hide the cord: the operator can show or hide the cord to switch between the paradigm with the cord and the one with just the avatar in front (2007 paradigm).
- Increase/decrease the cord’s thickness: the operator or the system can modify the cord’s diameter to show it thinner or thicker.
- Increase/decrease the cord’s length: the operator or the system can modify the cord’s length changing thus the distance between both avatars.

**Launch threatening action:** the operator can trigger it but it shall be triggered by the system at a specific time during an experiment. When triggered, a threatening action still not defined has to happen in the virtual scene making the subject react in a way that uncovers their emotional state.

**Use tapping:** used by either the system or the operator, this use case contains three main sub-use cases.

- Start/stop tapping: the actors can start the virtual tapping or stop it.
- Set sync/asynch: the tapping events can be put into synchronous mode (the default one) or asynchronous mode (which can be used as a control condition as explained in 6.2).
Virtual reality system to study human body perception

- Change tapping step: when the tapping is running, the object delivering the tapping shall be moving along several tapping regions on the avatar’s body. The system shall be in charge of controlling the advance of the tapping throughout the tapping path. The operator shall also be able to move the tapping device back and forth for testing or debugging purposes.

4.3. Software Design

The system we use in this study contains many different elements that work together to generate the virtual experience. There are some core components that are built using the XVR integrated development environment (IDE). Other components are required to interface with the network to send or receive data. Finally there is also a group of components that are required to communicate with VR and physio devices.

The XVR framework is a versatile IDE developed by a spin-off company from the Italian university Scuola Sant Anna. It has been developed to allow for fast integration of VR hardware and software components by means of its C-like scripting language. One of its main strengths is its scalability. It is able to render a VR project on media that range from HMD and PowerWall devices to complex VR installations such as CAVE systems or even an ActiveX component for the Internet.

The following is a diagram that shows the overall picture of our system.
Virtual reality system to study human body perception

Figure 8. High-level view of the different components integrated in our system
Virtual reality system to study human body perception

The core components of our system are the XVR functions and classes (Green). The XVR component is responsible for most of the functionality built into this project. This module performs tasks like the rendering, animations and graphic transformations.

To make possible an experiment in a virtual environment, we also need to communicate with other devices and to store some data to analyze in a later stage. Our XVR application receives data from two tracking systems. On one side, it gets data from an InterSense tracking system. A VRPN\(^3\) [15] server (red) gets the tracking data from the InterSense system. Eventually, a collector (grey) reads that data and forwards it to the XVR software. On the other side, the XVR software also has to read data from an Optitrack tracking system. We also use the collector (grey) to read tracking data from the Optitrack VRPN server and forward it to the XVR application.

Our application uses all this tracking data to apply transformations to the avatars and to the camera, which works as a window for subjects to observe the virtual environment. Thanks to the real-time tracking data we can provide subjects with visuo-motor contingencies to enhance Pi and full body ownership.

In addition to tracking data, our XVR application also reads data from a physio device. This device, which is a Nexus-4 model, sends some physio data to a computer via Bluetooth. The data is buffered by the PraxisPhysio server (red). The NexusXVRClient client connects to that server to read the data and send it the XVR application.

Another important component in VR projects that have been designed to run experiments is the Event Server (red). The event server allows us to send event IDs from our application whenever something significant happens during the experiment. Usually studies analyze data gathered during the experiment taking into account those events that were recorded during the course of the experiment. If a specific event happened at time \(t\), it may be interesting for the researcher to analyze the data in the interval \([t - t1]\) and \([t + t2]\) to study how data evolved in the time window \(t1 + t2\) before and after the event.

Aside from all the data input that the XVR application accepts, there is another essential part of the project where all the important data necessary for the statistical analysis needs to be stored. The element in charge of gathering these data and writing them to a file is the PraxisPhysioTrackerAndEventClient client. This component acts as a client for 3 different

____________________________\(^3\) VRPN stands for Virtual Reality Peripheral Network and it is a free, open source tool developed to ease the communication with VR devices. It acts as a device server.
servers. It connects to the VRPN server to get the tracking data from the InterSense device. It connects also to the events server to get data related to the events that were sent from the XVR application to be stored. And it finally connects to the PraxisPhysio server to get physio related data. This triple client connects to each one of those servers, gets data from their buffers and stores them in three different files to keep tracking, events and physio data separate.

The XVR software project (green color in the diagram) is organized in 3 layers. The first layer allows us to interact with the software in run-time. The second layer performs most of the computations and is in charge of updating animations, drawing the graphics context on the OpenGL viewport and updating any element necessary to run the project. The third layer is in charge of storing some data we want to be persistent. Every layer is described in more detail in the following paragraphs.

The **first layer** is the user interface (UI) layer. It is different from the typical UIs found in most software packages in the sense that there are no toolbars, buttons or drop-down menus. Instead, our UI has a few keys that trigger specific functions and a tracking system that allows subjects to interact with the environment as they move their body by means of specialized tracking hardware. The system displays a head-up display (HUD) showing the available commands. Also part of the graphic user interface (GUI) is the OpenGL viewport that is seen in monoscopic mode by a user and in stereoscopic mode by subjects wearing an HMD.

The **second layer** contains around 90% of the code of the project and is in charge of running most of the features that are required by the project. It controls animations for the tapping system, the avatars and other elements in the scene. It is in charge of setting up everything in the scene, calculating and updating the graphics to be rendered in the OpenGL viewport.

The **third layer** is in charge of the persistence. Data that need to be saved or read from the disk go through this layer. Data that need to be persistent are tapping system data, physio data recorded during the experiment, head tracking data and the timing of any events triggered during an experiment so that we can correlate them with the physio data.

### 4.3.1. The Avatars

Avatars are controlled by means of the Hardware Accelerated Library for Character Animation (HALCA) developed by Bernhard Spanlang [16]. The diagram in Figure 9 extracted from [17] describes the HALCA architecture.
HALCA works on top of Cal3D. Cal3D is an open source 3D character animation library based on skeletons [18].

By using HALCA we can import avatars with a skeleton into an XVR project and access the joints and bounding data for each bone. It gives us access to the avatar’s skeleton so that we can animate the different bones or apply all sorts of transformations to them.

HALCA is in charge of the skeleton initialization and in charge of any animations we wish to confer onto the avatar. This tool also gives us the possibility to specify morph targets in order to create procedural animations at a skin level, that is, by moving vertices. This functionality is performed through shaders. HALCA allows us to use shaders to transform the skin vertices and to apply changes to the surface appearance.

In our project, we use HALCA to animate our avatars and also to animate the cord. The cord has been rigged with an internal skeleton so that we can animate it in real time in the XVR project through HALCA.

### 4.3.1.1. Upper Body Part IK

We believe that visuo-motor contingencies are a key feature to induce some sense of ownership in VR experiences. For that reason we want subjects to move their virtual body as much as possible as if it was their own body.

Ideally their virtual body should move in the same way as their own physical body moves. Nowadays, that is still difficult to achieve with a high level of accuracy on a full body without having to put a full body tracking suit on subjects. Markerless tracking systems are quickly
Virtual reality system to study human body perception

progressing—Microsoft’s Kinect system is a good example—but are still far from the precision achieved with the systems that track special markers.

In our project, it was sufficient to have full motion for the upper part of subjects’ body. As we already pointed out in the general non-functional requirements listed in 4.1.2, subjects are not allowed to walk in the VR scene. For this reason, and to keep the experiment setup as simple as possible, we do not use a full body tracking suit in our study.

To take advantage of the visuo-motor contingencies we allow subjects to freely move the upper part of their body and we apply all those movements on the avatar. For instance, when subjects bend their spine to the right, they will see the avatar’s spine bend to the right too.
When subjects look to their left, the avatar’s head turns to the left just as subjects’ head must do if they want to change their visual perspective to the left. And of course, these changes occur in real time.

The system mimics subjects’ upper body part movements closely in a very simple fashion and using only one tracking device. We compute three different transformations for the upper body part:

- We take the position and rotation of the tracker placed on the HMD to compute the position and rotation of the virtual cameras that serve as the subjects’ eyes in the VR scene.
- We use the HMD tracker rotation in 3 axes to compute the avatar’s head yaw, roll and pitch. This way, when subjects bend their neck around any of the three axes, the avatar’s head will be doing the same movement in VR.
- We also take the HMD tracker position to compute the direction and the angle that the avatar’s spine has to bend. To properly bend the spine of the avatar we developed a simple IK (Inverse Kinematics) algorithm to bend the spine in a realistic and anatomically correct way.

An initial calibration is required to adjust the IK algorithm at the beginning of the experiment. Subjects have to be standing and looking straight ahead. The position of the HMD tracker is stored and used as a reference for any computations performed by the upper body IK algorithm. We ask subjects to remain still and to look straight ahead and then the operator presses a key in order to do the calibration.

More details on the implementation of the IK can be found in 4.4.
4.3.1.2. Arms IK

In addition to the spine and the head, the arms play a very important role in body self-recognition and body ownership. On a first primitive level of ownership, we have body agency. That is, we feel we can control our virtual body with our own movements.

We allow subjects to move the virtual arms of their avatar by moving their own arms. We do this by adding a tracker on each hand and then estimating the whole arm position using an IK algorithm that is enclosed in a .dll file. Our project makes use of several modules that are imported as .dll files. The IK for the arms is one of them.

To use the IK algorithm for the arms, we need to perform an initial calibration at the beginning of the experiment. The calibration consists of storing the initial 3D positions in the VR scene of the arms trackers. Then, we pass those values to the IK algorithm that computes the arms joint rotations. To perform the calibration we ask subjects to leave their arms still hanging on both sides of their body in a neutral position. The operator then presses a button to store the tracker values.

There are some limitations when using the IK on avatars’ arms. With this IK algorithm we only can achieve a limited degree of precision since the systems know nothing about the position of the elbow or the rotation angles on every joint. This precision is good enough though for this application since it allows us to have arms tracking with very little hardware setup.

Another known limitation is that the algorithm does not take into account hand rotations. So, if we rotate our hands around our wrist, there is no rotation on the avatar’s arms. Only tracking devices positions are used to compute the arm’s new configuration.

4.3.2. The Cord

The cord is one of the most complex elements in our VR scene. We model the cord as another avatar. This way, we can animate it and access its joints, as we would do with any avatar, through the HALCA libraries. At the same time, we apply several effects on the cord surface by using GPU programming.

We wanted the cord to behave as a believable connection between the two avatars—the one collocated with the subject and the other one in front—. To achieve that, the cord has the capacity to react to some events occurring in the environment.

One of the main events is the tapping. Tapping happens (visually) on the avatars and also on the cord itself. We are interested in giving the cord a realistic and believable appearance.
Virtual reality system to study human body perception

When the tapping object is touching the cord surface, we apply a wobbling effect on the surface so that it seems it was actually tapped. The effect is performed by a GLSL shader that computes the ripples originating at the collision point between the tapping ball and the cord surface.

Another event that can occur is the subjects’ own movement. Since the cord is virtually attached to subjects’ belly, their body movements should affect the cord. The shader is also responsible for applying a slight waving movement all the way along the cord when the subject moves abruptly.

There is still another event happening in the physical world that we would like to have an effect on the cord to strengthen the link between what is real and what is virtual. We are talking about subject’s breathing. The cord’s shader is also responsible for changing its radius according to subjects’ breathing. By using a respiration sensor attached to the Nexus-4 we detect when subjects are inhaling and then we can increase the cord’s thickness by a factor proportional to the inhalation. This way, during the experiment, subjects are breathing and looking at a cord that “breaths” just like them. The cord changes its size in a similar fashion as their belly. When their belly would grow to inhale, the cord increases its radius to simulate the breathing action.

Also, one of the requirements for the cord is that the operator should be able to modify its length and thickness in real time to try the experiment in different conditions until he finds the one that works the best with subjects. Again, we use a GLSL shader that increases the thickness and the length of the cord.

The shader changes the cord’s length by multiplying its coordinates by a scale factor in the Y axis, since the cord is aligned along that axis. To change the thickness, the shader moves every coordinate along its normal in the XZ plane, which is the radial direction.

4.3.3. The Tweener

It is often necessary in VR projects to create short animations to draw subjects’ attention, to play a task or just to interact with subjects. For example, in this project we need to animate the object we are using for the threatening final action at the last stage of our experiments and we need to animate the virtual object that is going to be used for the tapping.

Instead of designing some functions to just animate those objects in the VR scene with non-reusable, custom written code for our study, it was decided to write a simple animation.
Virtual reality system to study human body perception

system using several classes that everyone could use in future VR studies. We call such a system *The Tweener*.

The Tweener is a very basic animation package that allows us to animate several objects’ properties such as position, rotation, scale or colour. The animations we can program with the Tweener are simple *keyframe-to-keyframe* animations. That is, the animation is based on just two keyframes that store the initial and the final value of a given property. The Tweener system is responsible for creating the *tweening* [19], that is, all the intermediate frames between the first and the last value of a property to give the appearance that the object is evolving smoothly from one point to the other.

Moreover, the Tweener can produce smooth animations that transition from the start to the end in a variety of ways. The animation can start and/or end with constant speed, accelerating or decelerating. Any of the possible combinations can be played.

The Tweener can also play animations in three modes: *once*, *loop* and *ping-pong*. When the Tweener plays the animation in *once* mode, the animation is played just once from the start to the end. In *loop* mode, the animation is played from the start to the end a number of times. In *ping-pong* mode, the animation is also played a number of times alternating between the start-to-end and end-to-start cycles.

The following UML class diagram illustrates the objects design.
4.3.4. Tapping

We call “tapping” the act of slightly touching something or somebody repeatedly and randomly to stimulate the tactile system. It is commonly used in experiments to draw the attention of subjects’ visuo-tactile system to a certain part of their body in order to elicit...
partial or full body ownership illusions. For instance, it was used to induce the illusion of owning a rubber hand, as explained in 2.2. The object usually used for the tapping is a ball or a similar rounded object being about the size of a tennis ball.

In our project, tapping can be done in two ways. In the manual mode a person holding a tracked wand with a soft ball attached to it performs the tapping. A virtual tapping ball moves according to the tracked wand movements. In the automatic mode, the virtual tapping ball moves according to some algorithms described in 4.3.4.2 while the real physical tapping object is replaced by a haptic system which is described in 4.6.

Manual tapping has some advantages but several limitations. While manual tapping provides us with full control to apply a very irregular and randomized stroking to subjects, it is hard to parameterize and control. In other words, it is not easy to apply similar tapping sequences to different subjects.

There are also some precision problems as subjects may have different body sizes resulting in the virtual ball sometimes not reaching the surface of the avatar’s body and other times moving too much beyond the virtual body limits, potentially breaking the illusion.

As an alternative we can use automated virtual tapping—and occurs only inside the virtual world—. Hence, it is easy to control and adjust.

After performing manual tapping on subjects for several weeks, it was decided to implement a fully automated virtual tapping to be able to perform tapping without human intervention.

4.3.4.1. Manual tapping
A second person does the tapping on the subject. In order to do this, a VR device called “wand” can be used. This wand is part of the equipment that comes with the InterSense IS-900[13] tracking system and allows us to interact with the virtual content. The wand is a 6 degrees of freedom device that provides translation and rotation information to the tracking system. It has some built-in buttons we can use to send commands to the VR system.
In particular, we use the wand to move the tapping object. We attach a soft ball to the wand sensor. That is the ball that will be physically touching the participants’ body during the manual tapping.

The tapping process starts on subjects’ chest where they would feel a ball tapping their chest and would see the virtual ball doing just exactly what they feel.

At some point, that real tapping moves slowly down to the belly. Subjects see the tapping moving smoothly along the cord, as if the cord was just another extension of their body, an additional limb. The tracked ball movement is interpolated and used to move the virtual ball along the cord. This interpolation is performed using a circumference equation. The algorithm takes the position of three cord bones and uses them to calculate the circumference equation.

As the virtual tapping reaches the second avatar back, the real tapping starts fading out on the belly to fade in the real subject’s back. At that point, subjects feel a tapping on their back, while they see a virtual tapping on the back of the avatar in front. This is done by adding an offset to the virtual tapping ball in the corresponding axis.

4.3.4.2. Automated tapping

In this case, an algorithm that moves the tapping object along an absolute path performs the tapping. This path can be specified interactively in the 3D scene. The operator has to define a sequence of tapping points or steps. Two points define each step. The first point stores the initial position of the tapping object. The second point represents the actual tapping point.

For instance, the smallest tapping path would be a 1-step path defined by just two points. The tapping object would then be animated in between those two points following a pingpong movement. Figure 12 shows a more elaborate tapping path used to apply tapping on the two avatars and the cord.
The operator decides the tapping timings by specifying an array with several values, each value corresponding to the number of seconds the ball has to keep tapping in that a position.

The automated tapping engine (ATE) uses the XVR Tweener (4.3.3) to move the tapping object in a non-linear way. This makes the tapping more human-like when changing the speed of the object to perform the tapping.

To make the automated tapping sequence unpredictable by subjects, the ATE keeps randomizing the speed and the tapping distance. Initially, the ATE follows the path defined by the automated tapping user. But the ATE also moves the tapping randomly to the next or previous tapping step from the tapping step currently playing in the sequence defined by the user. In other words, the ATE always follows the tapping sequence defined by the user, but it randomly adds an offset of one or two tapping steps, forward or backward, in relation to the current tapping step in the sequence.

Figure 13 illustrates the UML class design for the automated tapping system.
Virtual reality system to study human body perception

4.3.5. Shaders

We developed a number of shaders to create several special effects in the VR scene. We decided to use shaders because it is a relatively new technology growing quickly and it is extremely powerful when it comes to generating graphic operations and modifications in real time.

In this project, we use shaders to fake shadows, modify geometry, generate reflections and improve textures.

The objects we render through a shader are:

- The avatars: we use a shader to change the geometry and to fake shadows coming from the tapping ball.

Figure 13. UML class diagram for the automatic tapping system
• The cord: its shader also can change the cord’s geometry and generates a shadow according to the distance from the tapping ball to the cord’s surface. This shader also applies a normal mapping effect to improve the final look of the cord.
• Fire extinguisher: the shader generates a reflection of this object.
• Trash can: we use the same shader as for the fire extinguisher. We also render a reflection for this object.
• Emergency door: again we use the reflections shader to create a reflection for the door.

4.3.5.1. Shadows

Shadows in VR experiments are not a must-have feature in general. Nevertheless, it has been shown in recent studies that visual realism in virtual environments that use global illumination, shadows and reflections improves the realistic response from subjects [20] [21]. In our study, we think it may be good to have shadows to enhance some of the events that subjects need to properly understand and pay attention to.

Shadows are helpful in computer graphics when it comes to perceiving distances among VR objects. During the tapping, collisions are constantly happening between the tapping ball and the objects of the environment. Actually, the tapping ball touches many times the surface of the avatars and the cord. We think that displaying a shadow for the tapping object may enhance the Psi factor in the sense that subjects may actually perceive the tapping as a more believable event that is actually happening and touching some other surfaces in the VR scene.

![Figure 14. Shaders add shadows for the tapping ball on the cord and avatars](image)

4.3.5.2. Geometric Distortions

We use a shader to change the length and radius of the cord in run time. We perform those changes in the vertex shader. We need to pass on to the shader the value of the variables that are going to control how much are we applying the transformation.
HALCA supports geometry distortions by using morph targets. We use a shader to morph between to different versions of the same mesh to produce the effect of breathing. The vertex shader moves a group of vertices from the original shader to the position of those vertices in the target mesh. This is used to increase and decrease the volume of the avatars’ chest so that they look like they are breathing.

4.3.5.3. Normal Mapping
Normal mapping is a 3D render technique that allows us to create the illusion of an object having much greater detail than it actually has. With this technique we can just use low-polygon count objects in the scene and render a texture for them that emulates some additional fine geometry on their surface. We create the illusion of a geometrically complex object while keeping the total number of polygons in the scene low.

We use normal mapping to add more complexity to the cord. A high-resolution model of the cord was developed initially. The model’s mesh was optimized to have fewer polygons before exporting it to XVR. Since we have the hi-res and low-res version of the cord model, we can generate in 3D Studio Max the normal map that will be used by the shader to create the normal mapping effect.

The fragment shader performs the calculations in tangent space to create the visual effect.

4.3.5.4. Reflections
As mentioned in 4.3.5.1, reflections play an important role when it comes to respond in a realistic way in a virtual environment. Reflections have been found to be important to enhance Psi. For this reason, we also developed a shader that renders object reflections on the floor.

For every object for which a reflection is required we have to render it through this shader. This shader computes the reflected geometry and its attenuation, as it gets further away from the floor level. Figure 15 shows the final result for a couple objects in the scene.
4.3.6. User Interface

Two classes are mainly involved in this layer. The first class is the `keyboardManager`. This class is in charge of keeping track of any key the operator presses. Once a key is pressed, this class calls the corresponding callback function. This class is also in charge of registering every command with a second class responsible for displaying the HUD.

The second class is the HUD and its main function is to display a menu with available options on a screen overlay. This class displays a 2D transparent rectangle over the 3D scene with a list of available commands.
Virtual reality system to study human body perception

The user of those classes can register new commands through the keyboardManager. The keyboardManager will register then the command with the HUD class. Each command belongs to a menu. Each menu is like a commands category. With the use of menus we can nest commands at different levels so that they are all better organized. Both classes know the menu that corresponds to each command.

The keyboardManager will only trigger the commands present in the currently active menu. The HUD will only display the list of commands that are included in the menu currently selected. So both classes work together to create a very simple user interface where the user can check the commands available in a screen overlay that can display different menus and execute functions by pressing keys.

The following UML diagram illustrates how both classes are connected to the main applications.

![UML diagram](image)

Figure 17. UML diagram class for the user interface

4.4. Software Implementation

4.4.1. The Cord

The cord’s shader increases the length of the cord by multiplying its coordinates by the identity matrix with the Y scale factor set to a value different than 1.

It is possible to modify the radius of the cord in execution time. This is possible thanks to the shader responsible for the cord rendering. The shader takes each vertex and adds a displacement in the normal direction. That is the reason why the cord surface should be as smooth as possible. Otherwise, vertex normals would point into a direction very different from
the radial one and the radius change would distort the cord too much. The radius factor is a variable that the XVR application passes to the shader as a uniform float variable.

It is also possible to change the length of the cord in execution time using the keyboard. The shader performs a scale change along the Y axis of the scene—and the cord is also aligned with this axis—by multiplying each vertex coordinates by a scale matrix. The length factor is also a variable that the XVR application passes to the shader as type uniform float.

The cord reacts to subject’s movements. If subjects move the upper part of their body, the cord waves to give visual feedback to the subject. The wave is generated by means of a damped harmonic oscillator equation [22] with the following pattern

\[ x = A e^{-\gamma t} \cos(\omega_1 t + \phi) \]

The wave’s amplitude is a function of \( t \). The wave is damped by the exponential term

\[ \gamma = \frac{b}{2m} \]

where \( b \) is the dumping force intensity and \( m \) is the oscillating mass. It is easy to adjust \( b \) or \( m \) to change the dampening behavior.

We need another wave equation to model how big the waves are along the cord. There has to be no movement on both bounds, as they are attached to avatars. The wave used to model oscillations along the cord maximizes vibration over the middle and minimizes it when vertices are closer to any of the two bounds. There is no movement at all for vertices located at the very beginning or end of the cord.

![Figure 18. Waves' amplitude along the cord is modeled by means of another wave to avoid any movement on both bounds of the cord. This way the cord looks like it is really attached to the avatars](image)
Another visual feedback consists of some concentric waves that ripple when the virtual tapping ball collides with the cord surface. The collision point is detected by comparing the tapping ball position passed onto the vertex shader with every vertex coordinate. The oscillation equations used to create these waves are essentially the same ones used to generate waves along the cord. Also a dampening effect has been added so that rippling decrease over time. To create the rippling effect, we apply the same oscillation phase to all the vertices at the same distance from the collision point.

4.4.2. The Avatars

4.4.2.1. Upper body Part IK
Quaternions are used to implement the IK algorithms for the upper body part. We decided to use quaternions because of the many advantages they have compared to other rotation techniques like matrices or Euler angles. We use quaternions to compute the rotations for the avatar’s head and for the avatar’s spine to properly bend it according to subject’s movements.

To achieve head rotations in sync with subjects’ head movement we get the three main rotations (yaw, roll and pitch) from the camera—which moves and rotates according to the head tracker—and compute the quaternions we need to apply to the bone of the head.

To compute the IK for the spine we need to build the quaternion we will apply to some spine bones. To build the quaternion we first calculate the angle between the reference vector, which goes from the spine bone to the position of the head in the calibration stage, and the target vector, which goes from the spine bone to the head tracker position. We also compute the axis of rotation from the cross product between the reference and the target vector. The following illustration shows the reference and target vectors as well as the avatar’s skeleton system with bone names.
Once we have the angle and the rotation axis we can build the quaternion to be applied to the spine. To keep an anatomically correct bending of the spine we distribute the rotation between two bones: spine1 and spine2.

4.4.2.2. Arms IK

The IK for the arms is a black box for us. We provide the position of the hand trackers and a DLL computes a possible solution for the joint positions and rotations for both arms.

4.4.3. The Tweener

According to the Wikipedia,

“Inbetweening or tweening is the process of generating intermediate frames between two images to give the appearance that the first image evolves smoothly into the second image.”

At present, the Tweener can animate any combination of the following object parameters:

- Position
- Rotation (angle and axis)
- Scale (axis independent too)
Virtual reality system to study human body perception

- Color
- Alpha
- ColorText
- CamDirection
- Callback (user custom parameter)

Usually, we do not find pure linear movements in real world. That is the reason why we have implemented several kinds of tween animations to generate different effects. These types of tweens are called *easing types*, and they are responsible for creating animations at different speeds and accelerations increasing the realism of movements. So far, the easing types implemented in the Tweener are the following ones:

**Linear (no easing)**

**Regular.EaseIn cubic**

**Regular.EaseOut cubic**

**Regular.EaseIn cubic**

**Bounce.EaseOut**

**Parabolic**

**Callback (user custom easing type)**

In our Tweener we implemented the easing equations described by Robert Penner [23]. There are many more easing equations and it is very simple to add them to the Tweener to have new kinds of animations.
Virtual reality system to study human body perception

Every easing type can also be combined in the Tweener with any of the three available playback modes described in 4.3.3: once, loop and pingpong.

The Tweener inherits from a class called TimeTable, which is in charge of storing the events we register—tweens and other kinds of events—, and the time when they should be triggered. The TimeTable works essentially as a timeline where we can register events to be triggered some time in the future.

The Tweener has also been integrated with a .dll that works as an event server. The event server stores events that were triggered within the application and that are usually relevant for the study. These events are eventually read by the client responsible to write them in a file for the statistical analysis.

To register an event in the server we just need to pass the ID of the event to the Tweener at the moment of registering the tween or callback that will trigger that event. When the Timetable triggers the event, it also sends the event data—event ID and timing—to the events server. The next figure shows a sequence diagram illustrating how events are registered in the Timetable and sent to the CVmEventServer instance object.

![UML sequence diagram](image)

**Figure 20.** UML sequence diagram explaining the main calls involved when sending events to the event server
Virtual reality system to study human body perception

Thanks to the event server we know the exact timing of the events in relation to the rest of the experiment. That is necessary to properly analyze physio data after experiments are finished. We are usually interested in looking for changes in physio data right after an event happened in the VR experiment.

We think we have made it really easy for users to add their own animations to a VR project. Once the Tweener is integrated in a project, adding an animation for an object requires only a single line of code. For instance, the following code would add a new tween to the ball object oBall, to change its position from [0, 20, 0] to [0, 0, 0] in 1.0 second, and that would be repeated over 3 times from the first point to the second one in loop mode. Also, that animation would start 3.0 seconds after the tween is registered for the object oBall.

```javascript
oTweener.addNewTween(oBall, "_position", "Bounce.EaseOut","loop", 3.0, [{0,20,0}, [0, 0, 0]}, 3.0, 1.0, event_id);
```

4.4.4. Tapping

In our study, the action of a ball tapping the chest, belly and back of subjects has to be mapped to tapping on the virtual chest of Av1, tapping on the cord and tapping on the back of Av2.

![Figure 21. Mapping between the visual virtual tapping and the tactile haptics](image)

In Figure 21 we show the mapping we need to create to show subjects a visual stimuli while providing tactile stimuli in a different region of their body that does not necessarily have to match the body part that they see it is being stimulated.

To achieve that mapping we implemented two different strategies: manual tapping and automatic tapping.
Virtual reality system to study human body perception

4.4.4.1. Manual tapping

The manual tapping is the tapping performed by a person standing next to the subject and holding the InterSense Wand with a soft ball attached to it. While the person taps a subject’s body, the virtual tapping has to be mapped on a region of the virtual space different than that of the physical one.

Initially we have to perform a calibration of the InterSense Wand device to add the appropriate offsets that account for the difference between the coordinates system of the XVR scene and the one used by the InterSense tracker. That is enough to perform the tapping on the area A.

For the area B, corresponding to the cord, we need to apply the real tapping on subjects’ belly while showing visual tapping on the cord. We achieve that by distorting the virtual space around the belly so that it corresponds to a circumference placed on top of the cord. The following figure explains this mapping graphically.

![Figure 22](image.png)

Figure 22. The image on the left shows the real tapping being applied to the subject's belly. The image on the right shows the circular mapping implemented to do real-time manual tapping on the cord.

Whenever we apply tapping on the belly, the virtual ball movements are translated onto the cord following the equation of a circumference. We use the parametric representation of the circumference

\[\begin{align*}
  x &= r \cdot \cos \theta \\
  y &= r \cdot \sin \theta
\end{align*}\]

where \(r\) is the radius and \(\theta\) is the angle respect to a reference angle we compute in the initial calibration. We only need the 2D representation of the circumference because the cord bones
are contained in the plane YZ. So we can actually calculate the circumference equation ignoring the X axis.

The circumference equation is computed using 3 points that correspond to 3 bones of the cord: 4, 6 and 9. The equation also takes into account changes on the radius of the cord—created by the corresponding cord shader—to change the radius of the computed circumference.

Finally, for the area C, we just need to translate all tapping movements happening on the back of subjects to the back of the avatar in front.

### 4.4.4.2. Automatic tapping

With the automatic tapping, the visuo-tactile remapping is easier to control. We have to setup the virtual tapping path in the XVR project. We then register the correspondence of every tapping step with every vibrator. When the ball is tapping on the ith step in a tapping path with 1 to n tapping steps, we trigger the ith vibrator from a set of n vibrators attached to the haptic vest. We establish a one to one correspondence between every tapping step and every vibrator. It could be done in a different way if we were to look at other applications, like establishing a one-to-many or many-to-one correspondence between tapping points and vibrators to study different haptic mappings.

The automatic tapping is implemented using the tweener classes. In fact, the automatic tapping combines two orthogonal groups of tweens to create the tapping movement. On one hand, the tapping object has a position tween in pingpong mode that makes the object move between two given dummy objects which are not rendered. On the other hand, these two dummy objects have position tweens applied to them to just move from one tapping step to the next one. The combination of these orthogonal tweens generates a natural movement for the tapping object. The following picture illustrates how tweens are combined to create the tapping animation.
Virtual reality system to study human body perception

Figure 23. A tween (blue) adds a pingpong movement to the tapping ball between two dummy objects. At the same time, another couple of tweens (red) move the dummy objects from one tapping step to the next one.

The automatic tapping has been implemented in a way that emulates the tapping that a person would perform. It can randomize different parameters like the tapping speed, the tapping distance from the initial point to the point where the tapping object actually touches the surface and the area being tapped.

Users can program a tapping sequence with 1 to n steps and can specify the times to switch from one step to the next one. Then the tapping usually will start on tapping step 1 and will go on through all the steps until it reaches the nth tapping step. Usually this tapping sequence has to be controlled in the experiment since we are interested in tapping first some areas of the body than others. Despite of the prerecorded sequence, the tapping system can locally randomize the tapping sequence. Let us say the tapping is happening on the ith step. The system can randomly move the tapping back to the step i-1 or to the step i+1 regardless of the sequence that the user programmed. This behavior is just local around the current tapping step and will be overridden by the tapping system itself when it is time to move to the following tapping step to advance in the sequence.

The main idea behind that random behavior is to illusory break the sequence to recreate the experience of a tapping being performed by a human instead of a machine.

The tapping system triggers the corresponding vibrator every time the tapping object collides with the point that was defined on the surface to be tapped. The collision detection is
Virtual reality system to study human body perception

performed between the tapping object and the dummy object that we place—but we render transparent—at the point that was defined on the surface to receive the tapping. There is no collision detection between the tapping object and the actual surface we are tapping. The collision detection between the two objects is performed by a function inside the XVR library. The following picture shows the different elements involved in this process.

![Figure 24. The tapping system triggers a vibrator every time a collision is detected between the tapping object (1) and one of the dummy objects (2) that we use to define the tapping zone on the surface of another object](image)

To use the automatic tapping system we only need to import the class `autoTapController`. This controller creates an instance of `autoTapPath` for every path we want to store and the `autoTapPath` class would create an instance of the class `autoTapStep` for every tapping step we create to build the tapping sequence. The whole process of creating a tapping path is interactive and allows us to quickly setup a path by creating a cloud of 3D points in the scene. The path information is stored in a file so that we can recover the last path we created upon restarting the project.

### 4.4.5. Shaders

We have implemented different vertex and fragment shaders to create the effects we needed in our VR scene. Some shaders were first developed using the shader profiling tool RenderMonkey™ from AMD. That tool helped us achieve the desired effect with shorter development cycles.
4.4.5.1. Shadows

We generate two kinds of shadows in our scene. First, we have a shadow for the tapping ball when it gets close enough to the cord and to the avatars. Second, we also generate the shadows that the cord and the avatars would project on the ground.

To compute the shadow corresponding to the tapping ball our XVR application passes the position of the tapping ball in the scene to the shader. The shader then computes the distance from the tapping ball to every vertex. If that distance is under a certain threshold value, we just darken the fragment according to the following equation:

$$ f(x) = \frac{x^2}{0.075^2} $$

The color of the fragment is then computed using the *mix* function from the GLSL library that returns a value computed as follows:

$$ a \cdot (1 - f) + b \cdot f $$

where $a$ is the black color, that is, $vec3(0.0, 0.0, 0.0)$, and $b$ is the color of the fragment computed from the texture and lighting for the object. The resulting effect is a shadow with a penumbra region.

![Figure 25. Shadow with penumbra](image)

We generate the shadow projected on the floor under the avatars and the cord by stretching a copy of these objects in a specific way. We set the Z coordinate for every vertex in these objects to the same value. We also paint every fragment for these objects black. The final result is that objects rendered through this shader become 2D black objects parallel to the floor with a silhouette that corresponds to the z-axis projection of the original object.
Virtual reality system to study human body perception

4.4.5.2. Geometric Distortions
We distort geometry through the use of shaders in two ways. For some effects we just translate vertices along their normal. That is how we achieve wave distortions and size changes on the cord, as explained in chapter 4.4.1.

For the breathing effects on the avatars, we use morph targets. In this case, the vertex shader interpolates linearly the position of every vertex between two values. One value comes from the position of the vertex in the original mesh and the other value comes from the position of the same vertex in the mesh that we pass as morph target.

4.4.5.3. Normal Mapping
As explained in chapter 4.3.5.3, we use normal mapping to increase the apparent level of detail on the cord’s surface without having to use a hi-poly model in our scene. To achieve that, we need to add a perturbation to every fragment normal according to the normal map we computed in 3D Studio Max. More information on how the normal map for the cord was generated can be found in chapter 4.5.2.3.

There are several normal mapping techniques. In our case we use tangent space to compute the perturbation that will change the original fragment normal. Tangent space is the most common technique and it has some advantages over the others.

The fragment shader starts by computing the 3 vectors that characterize the tangent space for a given fragment. Then it puts in tangent space coordinates the normal stored in the normal map. The resulting normal is added to the fragment normal that was passed on to the
fragment shader from the vertex shader to obtain the perturbed normal. Finally, the fragment shader computes the dot product between the light vector—calculated in the vertex shader—and the perturbed normal. The result of that product is then multiplied by the original color of the fragment to get the color components to be stored in the `gl_FragColor` variable.

### 4.4.5.4. Reflections

Whenever we want to add a reflection to an object in our scene we just need to render it again through this shader. In this case, the vertex shader is in charge of mirroring the geometry while the fragment shader is responsible for creating the light attenuation effect. Reflections typically have as we increase the distance to the floor.

To mirror the geometry we just change the sign of the Y coordinate for every vertex. That implies these reflections will only be correct on flat surfaces parallel to the XZ plane.

On the fragment shader side, we decrease a little bit the overall light of the reflected geometry and we add an attenuation factor so that the further away a fragment is from the floor, the more transparent it is according to the following linear expression:

\[ a(y) = 1.0 + \frac{y}{0.3} \]

So, the alpha channel \( a \) for every fragment is a function of the distance to the floor on the Y-axis. At the floor level, \( y \) is 0, so alpha would be 1.0 (100% opacity) for fragments. For fragments at distance 0.3 units under the ground, \( y \) takes the value -0.3 and then alpha is 0.0 meaning that fragments are rendered completely transparent. The value 0.3 was empirically chosen as a convenient value for the reflections we want to create in our scene.

![Figure 27. Reflections under the fire extinguisher and the door](image)
Virtual reality system to study human body perception

4.4.6. User Interface

The user interface has been implemented using the XVR library Console. This library has functions to draw overlaid 2D text using system fonts on the OpenGL viewport.

To implement the commands functionality we check for any pressed keys in the OnFrame() XVR loop. This loop is responsible for redrawing the OpenGL context for every frame.

4.5. 3D scene design

4.5.1. The avatars

Most of the work on the avatars required changes on textures and skeleton animation.

4.5.1.1. Modeling

The male avatar used in this project was bought by the EVENT Lab and comes from an avatars library developed by the company aXYZ Design. Only some small modifications on textures are required to meet our needs. Also some geometric modifications on the avatar’s skin are performed in real time. For instance, in order to generate the breathing effect, a shader moves a group of vertices on the avatar’s chest.

4.5.1.2. Texuring

Avatars already come with their own textures in the avatar’s library. The next figure shows the texture of the avatar we use in our scene.

![Texture used for the diffuse component of the material to shade our male avatar](image)

We have OpenGL’s backface culling turned off in our project since we prefer to see the polygons even when we place the camera inside objects like the cord or the avatar’s body
Virtual reality system to study human body perception

itself. Therefore, we have to do some modifications on the texture of the avatar that is collocated in subjects’ position. In order for subjects to be able to see the scene we need to make the head transparent. Otherwise the camera would only see the avatar’s face polygons facing towards inside the head.

![Figure 29. Texture used as an alpha channel for the avatar’s head](image)

The Figure 29 shows the texture we use as the alpha channel for the avatar’s head. In this case, black pixels are interpreted as 100% transparency while pure white pixels mean 100% opacity. With this texture, we only show the beginning of the avatar’s neck fading out to 100% transparency. The next image shows the final result.

![Figure 30. The neck fades out and the head is completely transparent](image)

4.5.1.3. Animation

Every avatar has a slight motion applied to the upper part of their body and to both arms. This motion was created from within the 3D application by animating some bones using keyframe-based animation. The animation mimics the movements we humans do when standing, when keeping our balance.

The animation lasts for a few seconds and is looped. The purpose of this animation is to break the perception that avatars are still like a rock. At the same time, the animation is very subtle.
Involuntary movements coming from your avatar would break the illusion that the avatar is under your control or it is yourself in the virtual environment.

4.5.2. The Cord

4.5.2.1. Skeleton

The cord is made of a mesh that is controlled by a 9-bone skeleton. The root of the skeleton hierarchy is right in the middle of the chain. This design decision was made to ease the cord inverse kinematics (IK) animation on both sides. In the case where we have the root placed on one bound of the skeleton, it is more difficult to animate that side. That is because when the root is translated, it drags with it the whole hierarchy. As each cord side is attached to an avatar, both sides have to be animated when the corresponding avatar moves.

![Cord skeleton](image)

**Figure 31.** Cord skeleton

4.5.2.2. Modeling

The cord skin has been modeled in 3D Studio MAX R2009. The modeling started with a cylinder with a high number of polygons. That cylinder had a noise modifier applied to give the cord a more organic look. We wanted the cord to resemble a real supersized version of an umbilical cord.

We initially modeled a high-resolution version of the cord. The model that is finally used in the virtual environment is a low-poly version of the original model. That was a requirement necessary to keep the total count of polygons in the scene as low as possible in order to speed up the stereo rendering on the HMD.

The original hi-res model and the low-poly model were used to create the normal map to be used by the bump mapping shader to render the cord’s skin.
Virtual reality system to study human body perception

4.5.2.3. Texturing

Basic texturing has been adjusted in a 3D modeling and animation package and an image editor. The basic texturing contains a noise texture and an opacity texture. XVR loads both textures using the configuration file for the cord.

Both textures have to be passed on to the shader responsible for some parts of the cord rendering. The cord fragment shader takes the noise texture and uses its RGB values to generate another noise texture using two colors hard-coded in the shader. That texture will be used as the diffuse component of the material for the cord. This would allow the shader to perform some noise animation to give the cord a more organic look.

The fragment shader also changes the opacity values of each fragment according to RGB values found in the opacity texture. Opacity is currently being used to smooth out the connection between the cord and the avatar.

![Figure 32. Cord bound smoothed out using the opacity channel](image)

In addition to that, normal mapping has also been added to the cord to increase the level of realism. The normal mapping effect uses a normal map, which was generated in the 3D software package using a model of the cord with higher LOD to have geometrical bumps spread all over its surface.
XVR loads the normal texture map through the cord configuration file. This texture is also passed on to the cord shader, which uses it to perform some perturbations on each fragment normal. As a result of that, we end having a cord that apparently has a high LOD with a certain amount of small bumps while geometrically keeping it simple and with normals pointing out radially from every cord section. As it will be explained in the animation section of this document, it is important not to distort very much the cord normals.

4.5.2.4. Animation

The cord model has been provided with a default looped animation similar to the one avatars have. Again, the purpose is like in 4.5.1.3 to add a very subtle random motion that breaks the rock-like stillness that virtual objects usually have. We intend to give the cord the appearance of an organic element rather than an inanimate object.
Virtual reality system to study human body perception

This animation was also recorded in 3D Studio Max using keyframes. It was exported as a Cal3D animation that HALCA plays when the XVR project is running.

4.5.3. The environment

The environment has been created in the modeling and animation software package 3D Studio Max R2009.

At a geometric level the environment is pretty simple. It consists of a room with a door and no windows. There are just three objects in the scene. Two of them are just **atrezzo**. They are a thrash can and a fire extinguisher. They are in the scene to just help subjects feel the experience a bit more plausible. The third object is a fan that is spinning on top of the first avatar’s head and that is being used as the final threatening action in the experiment.

![Figure 35. General view of the virtual room](image)

The environment and objects’ baked textures have been generated in 3D Studio Max R2009 with global illumination and ambient occlusion settings to increase the realism of the scene.
Virtual reality system to study human body perception

![Room floor with baked global illumination and ambient occlusion](image)

Figure 36. Room floor with baked global illumination and ambient occlusion

4.5.4. Animation

At the environment level, the only animation we have is the fan rotation. This animation has not been previously recorded. The fan is animated in execution time by the *tweener* described in 4.3.3.

Also, when the fan has to descend at the end of the experiment to threaten the head of the avatar in front of the subject, the translation is performed by the *tweener*.

4.5.5. Shadows

Shadows are an essential way to provide visual feedback to subjects. Shadows allow us to measure distances among objects.

Global illumination and ambient occlusion have been baked on textures from within 3D Studio Max. There are no projection shadows happening in real time in the scene. The cord and avatars shaders create the virtual ball shadow by changing the color of the closest fragments to the tapping ball to a darker color. Therefore, the only visible shadow in the scene is the one created by the ball on the cord and on the avatars’ surface.

Shadows projected by the avatars and the cord on the floor of the virtual room are also created by a shader that renders a copy of those objects with color and geometric changes so that they look like shadows.
4.5.6. Lighting

The current lighting implementation works with one single light source. It uses the fixed OpenGL pipeline for every object except for the two avatars and the cord, which are rendered through shaders.

The lighting implementation in the cord shader supports more than one light source. The lighting algorithm computes the light contribution for each source and adds it to some global variables for ambient, diffuse and specular components. [11]

At present only one light source is placed in the scene though. Apparently that is because the bump mapping only supports one light source for now.

4.6. Haptic technology

This project uses haptic technology in two different ways. On one side, the subject’s breathing is used to animate the cord radius at the same pace. On the other, the subject receives some vibrations depending on some events happening in the virtual world.

The breathing is controlled by means of a respiration sensor (NX-RSP1A) attached to a Nexus-4 device.

The breathing algorithm stores 30 maximum respiration values and 30 minimum respiration values. This algorithm detects these singular points by getting samples every 11 respiration values and looking for a change in the slope sign of the breathing curve. By getting samples every N values local singular points are avoided. At present, only a mean value is computed from the maximum and minimum values gathered. Nevertheless, a better filter like a Kalman filter could be applied to get rid of any outliers and improve the breathing algorithm. In the
Virtual reality system to study human body perception

final experiments the breathing device was never used because it stopped working properly before starting the first experiments.

The haptic vest with vibrators constitutes the main haptics system that subjects have contact with in this study. When the ball, following the automated tapping process, hits the virtual body, the subject receives a slight vibration on the corresponding area of the body.

The vibrators are connected to an Arduino board that we can program using Arduino software that allows us to write a firmware and upload it to the board via USB. With the Arduino board connected to a computer we can trigger the vibrators by writing a vibrator ID—usually a char—to the COM port the board is connected to.

![Figure 38. Front part of the haptic vest with several vibrators attached to it. The arduino board next to the vest connects every vibrator to the computer using a USB cable](image)

Each vibrator is meant to correlate the visual virtual tapping that participants see in the virtual environment with an actual tactile sensation. Vibrators will be triggered synchronously with the virtual tapping in the synchronous condition and asynchronously in the asynchronous condition.

When the virtual ball taps the cord, the subject receives a vibration in an area close to the center of the belly.
5. Pilots

It is necessary to test the whole experiment with some people before starting any real experiments. The pilots stage is meant to be a test to detect which parts of the project need to be modified, added or removed. It is in fact an iterative project that requires testing again the changes with new subjects. With pilots, researchers also get trained to be able to execute every part of the experiment smoothly and without introducing many procedure variations from subject to subject.

5.1. Design

The initial pilots were done using a different virtual environment that was completely designed to enhance the final virtual threat that subjects would experience after a few minutes of being in the virtual world.

The first virtual environment consisted of a completely different room similar to what we can find in adventure games like the classic Tomb Rider. The next figure illustrates that environment.

![Figure 39. Design of the old environment. The left image shows the scene at the beginning, before the ground moves down. The image on the right shows the front avatar standing right on the first isle very close to the edge](image)

At some point during the experiment, the floor goes down and leaves just some small isles. The avatar in front is left in a position very close to the edge. The threatening action consists in this case in showing subjects that the avatar in front loses its balance and is about to fall out. Our hypothesis is that if subjects feel some degree of ownership or embodiment with the avatar in front when it is about to fall, they will tend to correct the movement to avoid falling. As explained in the results 5.4, this does not work.

In another iteration, the threatening action of unbalancing was replaced by a ball with spikes that was launched against the body of the avatar in front of subjects. Again, our hypothesis is
that subjects will try to avoid being hit by the ball when they feel some sense of ownership with the virtual body in front.

In the last major iteration, the whole scene and the threat were replaced. The environment was replaced by the scene described in 4.5.3. The threat is then replaced by a spinning fan that comes down to the head of the avatar in front as if it was about to chop its head. This threatening action already showed its effectiveness in [24].

5.2. Questionnaires

The questionnaire used during pilots is much more simple than the one we use for the final experiments. In our study we ask a few questions and ask subjects about their experience to see what can we improve and what is not working at all.

The questions we asked were the following ones.

1. Did you feel ownership with avatar 1 (collocated avatar)?
2. Did you feel ownership with avatar 2 (avatar in front)?
3. What did you feel when the ground fell?
4. What did you feel when avatar 2 unbalanced?
5. What did you feel when the ball passed in front of you?

Aside from questionnaires, ECG data and head tracking data were also recorded. We never analyzed that data though.

5.3. Setup

More than 15 people were recruited for the pilots. The setup for each subject included several steps and in the following order:

1. 3 ECG electrodes attached to the skin
2. Breathing device described in 4.6 and Figure 37
3. Nexus-4 wireless device connected to the ECG electrodes and the breathing sensor
4. Haptic jacket with 10 vibrators attached to it. Described in 4.6
5. OptiTrack hand trackers
6. HMD with InterSense tracker for head tracking
5.4. Results

Pilots showed that the threatening action that showed the avatar in front losing its balance quickly broke the illusion of participants rather than eliciting the illusion of part of your self moving or falling. Participants reported the avatar in front was doing movements that had nothing to do with them.

Regarding the spiky ball being launched against the avatar in front, participants reported they were not sure if the ball could hit the avatar or not. They also said the ball was passing too fast in front of them. In general, they did not have the feeling of their body or the body in front being threatened.

As a result of that, the threat was replaced by the fan element. There were other minor changes in the design of the study. We added background sound to avoid subjects hearing the sound of the vibrators in the haptic vest. We also learned which were the right spots in people’s body to properly feel the vibrators. We decided to avoid placing vibrators in places were the human body has generally a pit (for instance, the middle part of the chest). In fact, we decided to use only males in our study. The fabric used for the haptic jacket is pretty rigid and it does not adapt very well to bodily curves.

It was at this stage that the breathing sensor broke. We decided it was not mandatory for the study, so we proceeded without showing any breathing feedback in the virtual scene.
6. Experiments

6.1. Ethics Approval

This study is part of a European project called TRAVERSE (Transcending Reality Activating Virtual Environment Responses through Sensory Enrichment). Every experiment we carry on at the EVENT Lab has been approved by an Ethics Committee.

6.2. Design

The experiment has two independent variables or factors. We have the type of paradigm and the tapping mode.

The following table summarizes the factors and their levels.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paradigm</td>
<td>Cord, No cord</td>
</tr>
<tr>
<td>Tapping</td>
<td>Synchronous, Asynchronous</td>
</tr>
</tbody>
</table>

Figure 40. The experiment has 2 variables with 2 values each

The variable “paradigm” accounts for the two different designs we are testing in the experiment. The “No cord” paradigm corresponds to [8]. While the “Cord” setup includes a co-located virtual body and a virtual connection between the two avatars in the form of an umbilical cord.

The variable “tapping” accounts for two different tapping modes: synchronous and asynchronous. The tapping in asynchronous mode is used as the control condition\(^4\) since it breaks the illusion of the virtual experience. The asynchronous tapping breaks the visuo-tactile correlation between the events that subjects see in the virtual environment and what they feel on their physical bodies. That ends reducing the plausibility. Subjects experiencing the asynchronous tapping will constitute the control group. The tapping in synchronous mode is the normal condition that gives subjects the full virtual experience.

\(^4\) It is very common during experiments to have a control group that acts as a baseline to help study thoroughly the effects on the group that receives the real treatment or experience. The control group is pretty much used as a reference.
Experiments

We want to study two paradigms. For each paradigm we have the normal group and the control group. Therefore the experiment will be run in four different conditions corresponding to every combination of the two independent variables. The following table describes each condition:

<table>
<thead>
<tr>
<th>Condition ID</th>
<th>Paradigm</th>
<th>Tapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No cord</td>
<td>Synchronous</td>
</tr>
<tr>
<td>2</td>
<td>No cord</td>
<td>Asynchronous</td>
</tr>
<tr>
<td>3</td>
<td>Cord</td>
<td>Synchronous</td>
</tr>
<tr>
<td>4</td>
<td>Cord</td>
<td>Asynchronous</td>
</tr>
</tbody>
</table>

Figure 41. In our experiment we test subjects in one of the four conditions. Each condition has been assigned an ID to ease the statistical analysis later on.

When it comes to decide which subjects do which conditions, in general there are two approaches to the design of an experiment: within-subjects and between-subjects. A within-subjects design is generally preferred for a number of reasons [25].

In our study there are several stages. The final stage consists of a threatening action that is presented to subjects. We use that action to measure subjects’ reactions. So, the surprise factor is critical for our measures. In our pilots, we tested 2 different conditions per subject. All subjects reported knowing that the final threatening action would happen again when they were experiencing our study for the second time in a different condition. Hence, there is a strong learning factor there. Carry-over effects would largely affect our data.

That is the reason why a pure between-subjects design has been chosen for our study. In that design, each subject performs one single condition consisting of one of the two paradigms and one tapping mode. We decided to work with 10 participants per group. In the pure between-groups design, each group performs one single condition. That means we need one group per condition for a total of 40 subjects.

<table>
<thead>
<tr>
<th>Paradigm</th>
<th>Tapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cord</td>
<td>10</td>
</tr>
<tr>
<td>No cord</td>
<td>10</td>
</tr>
</tbody>
</table>

Figure 42. The study is designed as a between subjects experiment with 10 subjects per group. Each group experiences one single condition to avoid carry-over effects.
Experiments

To avoid introducing a bias in our data due to ordering of conditions, we have to randomize the four groups. To do that, we use a web-based randomizer [26] that creates 10 sets with 4 conditions each without repetitions in each set.

<table>
<thead>
<tr>
<th>10 Sets of 4 Unique Numbers Per Set</th>
<th>Range: From 1 to 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set 1</td>
<td>Set 2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

Figure 43. This table shows the 4 conditions to be run in the experiment randomized throughout 40 subjects

6.3. Data Gathering

In every experiment we need to collect some data in order to analyze it later. This data allows us to understand what happened in the experiment. We need that data to run the statistical analysis that will eventually lead us to draw some conclusions, which is the final goal of the experiment.

6.3.1. Subjective Measures

One of the main sources of data in experiments is the use of questionnaires. They constitute and easy and cheap way to gather some information from subjects. Sometimes it may only be possible to get data through the use of questionnaires. That is the case of measures related to the ownership that someone feels about something. It is not easy to measure the ownership feeling and, when possible, it could require the use of expensive and bulky equipment like fMRI systems. For that matter, questionnaires bring us the possibility to measure ownership by simply asking subjects about their experience.

Since we intend to reproduce in our virtual reality system the results found in [8], our questionnaires are based on some questions already present in that study.

Also, regarding the answers, we use the same methodology used in the previous study.

Participants have to evaluate each question using a 5-point Likert scale. For each question, participants have to choose one of the following five options:

1. Strongly disagree
2. Disagree
3. Neither agree nor disagree
Participants are asked to answer two questionnaires after the experiment. The first one intends to capture participants’ impressions right after the experiment finishes, while they are still wearing the HMD and lights are off. Any short lasting feeling they had during the experiment should be easy to remember for them at this stage. The investigator reads out every question and participants have to say a number from 1 to 5 according to the Likert scale. The first questionnaire includes the following questions:

1. I felt as if the body I saw in front of me might be my body.
2. I had the feeling that I might be harmed when the fan descended over the avatar in front of me.
3. The movements I saw the body in front was making seemed to be my movements.
4. The body I saw in front was another person.
5. What do you think about the experience? How did you feel during the experiment?
6. I felt as if the cord was an extension of my body.
7. I felt as if I had two bodies.
8. I felt as if I was located at some point in the space in between the two virtual bodies.

The last three questions are reserved for the scenario with the virtual cord.

The second questionnaire is a more thoroughly one. It seeks to capture participants’ impressions with more accuracy, giving them the opportunity to think about their experience for a second time. They answer this questionnaire once lights have been turned on and all the VR equipment has been removed. Participants must answer those questions on a paper without the intervention of the investigator, except for doubts that may arise. The second questionnaire includes the following questions:

Q1 It seemed as if I were feeling the touch of the ball in the location where I saw the virtual body touched.

   a) The body in my own position
   b) The body in front
Experiments

Q2 It seemed as though the touch I felt was caused by the ball touching the virtual body.

   a) The body in my own position
   b) The body in front

Q3 It felt as if the virtual body was my body.

   a) The body in my own position
   b) The body in front

Q4 It felt as if my (real) body was drifting towards the front (towards the virtual body).

Q5 It seemed as if I might have had more than one body.

Q6 It seemed as if the touch I was feeling came from somewhere between my own body and the virtual body.

Q7 It appeared (visually) as if the virtual body was drifting backwards (towards the real body).

Q8 During the experiment the body I saw was that of another person

   a) The body in my own position
   b) The body in front

Q9 When the fan descended over the head of the virtual body in front, it felt as if it could chop my head.

Q10 When the ball touched the cord, did you feel that the touch was coming from where the ball touched the cord? If yes, indicate how far along the cord this happened.

Again, questions related to the cord are only presented to subjects that experience the conditions that include the virtual cord.

The full version of both questionnaires can be found on the appendices.

6.3.2. Objective Measures

Besides from questionnaires, two objective measures are also considered for this study. During the experiment, the head position of the avatar is recorded. A tracker mounted on top of the HMD provides that position. This gives a measure of any movements done by the subject during the course of the experiment to balance his or her body.

ECG data is also recorded for every subject to help find differences on the subject experience in relation to the baseline. ECG data is used later on to measure the heart rate (HR)
Experiments

deceleration of participants during the last stage of the VR experiment when a threatening action is presented to them.

It is been shown in recent studies that when subjects increase their attention to an event, they experiment HR deceleration.

For instance, according to [27], when subjects with snake or spider phobia are exposed to the elements that cause the phobia, they experiment HR acceleration. Subjects without those phobias show HR deceleration probably due to the increase of attentional resources.

6.4. Setup

This experiment uses two different tracking systems. On the one hand, an Intersense tracking system is used for the head tracking. A smooth head tracking is a key factor in the virtual experience as it is used to turn the camera around in the virtual environment following subjects’ head movements.

![InterSense IS-900 tracking system](image)

On the other hand, an Optitrack system is used to keep track of the subject hands. The setup consists of two passive trackers attached to the gloves that subjects will be wearing.

![Optitrack infrared camera](image)
At present, the EVENT Lab has two laboratories with a 12-camera OptiTrack setup. The cameras are placed in the *truss setup* configuration recommended by the OptiTrack manufacturer NaturalPoint.

![12-camera OptiTrack setup](image)

*Figure 46. 12-camera OptiTrack setup used in the experiments. Image courtesy of [14]*

Subjects will have 3 electrodes attached to their skin to record ECG data. The Nexus-4 device can measure ECG through 3 electrodes (positive, negative and ground).

![Nexus-4 three electrodes](image)

*Figure 47. Nexus-4 three electrodes to measure ECG data*

The negative and ground electrodes go on the chest and the positive one goes just under the lowest left rib. The following drawing shows how to set the Nexus-4 electrodes.
On top of the other devices, subjects will be wearing the haptic jacket with a set of 10 vibrators. The following figure shows a person wearing the haptic jacket and the HMD.

6.5. Stages

The whole experiment can be divided into 4 main stages.

On the first stage subjects are immersed into the virtual environment. Subjects will be asked to look around, move their upper body and arms. They have to get familiar with the new environment and their new body. It is also important that they play with their upper body movements to see how that affects the virtual scene. It is in this stage where visuo-motor contingencies can help participants achieve the sense of ownership with their virtual body.
Experiments

On the second stage subjects remain standing and are asked to focus on a ball tapping their body and cord. This stage goes on for about 5 minutes.

The actual experiment finishes with a threatening event unexpected by the subject. The spinning fan descends from the ceiling over the avatar in front of the participant.

After the experiment in the virtual environment, subjects are given some questionnaires with questions regarding their feelings and impressions in relation to what they just experienced.
7. **Statistical analysis**

The statistical analysis is a 2 by 2 factor analysis since we have two factors or independent variables with two levels each. We analyze the ECG and head tracking data that was recorded during the experiments to see if we can withdraw any objective conclusions about how subjects responded during the experience.

7.1. **ECG data analysis**

To analyze raw ECG data we mainly use MATLAB and a toolbox provided by g.tec called g.ECGtoolbox that can perform single beat classification and analysis.

Since we are looking for HR deceleration, as explained in 6.3.2, we are interested in analyzing times between heartbeats in the ECG signal. Three main parts that constitute the QRS complex characterize each heartbeat. The following figure shows a characterization of a single heartbeat from ECG data.

![Figure 50. Image showing a typical QRS complex. Image courtesy of Wikipedia [28]](image)

The g.ECGtoolbox MATLAB toolbox features a tool called g.BSanalyze that allows us to perform automatic QRS complex detection on ECG data. We have to load raw ECG data onto this tool and the complex detector function will automatically attempt to detect all QRS peaks present in the signal. The following snapshot shows some markers already set on every peak on the ECG data.
After we add markers to ECG data, we can proceed by analyzing them in MATLAB computing R-to-R (RR) time intervals to detect an acceleration or deceleration in the cardiac rhythm. An increase in the duration of RR intervals would mean that the subject experienced HR deceleration while a decrease in the duration of RR intervals would imply that there was HR acceleration.

We then analyze the RR intervals in a 10 seconds window right before the threatening event and we use that as a baseline to compare it to the analysis performed over another 10 seconds window that starts 5 seconds after the threatening event is triggered.

![Figure 51. g.BSanalyze displaying green markers on every detected QRS complex in the ECG signal](image)

![Figure 52. Timeline showing time windows we use to look for HR deceleration](image)
We do not take the 10 seconds window right after the fan starts descending because it takes around 5 seconds for the fan to get close enough to the avatar in front to enter the field of view of the HMD.

7.2. Head Tracking data analysis

In addition to analyzing the ECG data, we also perform an analysis on the head tracking data recorded during the experiments. We are interested in seeing if there are any significant differences in the total distance and the variance of the head movement before and after the threatening fan event.

Again, as we explained in the ECG data analysis in 7.1, we take a 10 seconds time window before the event and another 10 seconds time window after the event. We compute total distance and variance using MATLAB.

To compute the total distance first we compute the Euclidean distance between every two successive head tracking points. The final total distance is the sum of all point-to-point distances between successive points. In other words, given a set points

\[ \{p_1, \ldots, p_{i-1}, p_i, \ldots, p_n\} \]

with each point having coordinates

\[ p_i(x_i, y_i, z_i) \text{ with } i = 1..n \]

the total distance \( D \) for those points is

\[ D = \sum_{i=2}^{n} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \]

If we find significant differences between the total distance before and after the event, that may indicate that there were some behavioral differences during the experiment.

We are also interested in studying the differences in head tracking variability before and after the threatening event. Common variability measures are the variance and the standard deviation. To compute the variance on head tracking data we compute the average of the squared Euclidean distances from every tracking point to the mean point. In other words, given a set of points

\[ \{p_1, \ldots, p_{i-1}, p_i, \ldots, p_n\} \]
Statistical analysis

with each point having coordinates

\[ p_i(x_i, y_i, z_i) \text{ with } i = 1 \ldots n \]

and with their mean point being

\[ \mu(\mu_x, \mu_y, \mu_z) \]

the variance \( \sigma^2 \) can be computed as

\[
\sigma^2 = \frac{\sum_{i=1}^{n} (x_i - \mu_x)^2 + (y_i - \mu_y)^2 + (z_i - \mu_z)^2}{n}
\]

with \( n \) being the size of the sample we are analyzing. We can easily compute then the standard deviation by taking the square root of the variance.
8. Results

The results we present here come from a preliminary analysis on the questionnaires.

In the Lenggenhager paradigm, subjects reported having a higher sense of ownership with the virtual body in the synchronous condition than in the asynchronous one (see Q3b in appendix A).

In the cord paradigm, subjects gave higher answers in the asynchronous condition than in the synchronous one when asked about ownership feeling with their collocated body or the body in front. Subjects also reported higher ownership with the collocated body than the body in front in both conditions (see Q3a and Q3b in appendix A).

Also in the cord paradigm, subjects reported much higher punctuations when asked about feeling the tapping coming from the cord in the synchronous condition than in the asynchronous one (see Q10 in appendix A).

In the synchronous condition, subjects reported stronger ownership feeling with the avatar in front in the Lenggenhager paradigm.

When subjects were asked about the feeling that the fan could be chopping their head when it descended over the avatar in front (see Q9 in appendix A), they reported stronger results in the synchronous condition in the Lenggenhager paradigm and in the asynchronous condition in the cord paradigm. That is coherent with the fact that subjects experienced a higher degree of ownership with the body in front in the Lenggenhager paradigm, while they reported higher ownership with the collocated body in the synchronous condition with the virtual cord.

We will get more results in the coming weeks by thoroughly analyzing our questionnaires, physio and head tracking data.
9. Conclusions

On the technical side, we have developed a VR project that combines software, VR devices, haptic devices and physiology sensors. We have achieved all the non-functional requirements. We have also achieved all the functional requirements except for two of them that are related to the breathing. The breathing-related requirements were implemented and piloted but never used in the final experiments because the breathing sensor stopped working properly.

We developed a tweening animation system for XVR using object-oriented programming. This module has already been used in other VR projects proving to be easily reusable and scalable.

We also developed an automatic tapping system that allows for tapping in ways that were not possible when performed by humans. We showed for the first time that a fully virtual and automatic tapping system could be used to mimic the tapping performed by humans to provide subjects with tactile stimuli during a VR experiment.

We designed and developed functionality to provide subjects with visuo-motor contingencies inside a virtual environment. When subjects move the upper part of their body, they see their virtual body or bodies moving in the same way. This allows us to capture subjects’ movements with minimal tracking equipment. For more precision on the movements or full body movements a full body tracking suit should be used.

Shaders were used in this project to provide visual feedback in the form of waves and shadows when tactile stimulation was applied on virtual objects like the cord or the avatars, helping subjects “feel” the tapping in a more plausible way.

We also developed shaders to increase the level of realism of the VR experience by means of shadows and reflections.

Several pilots and 41 experiments were run using this software application and the required hardware. As a result of that, subjective data through the use of questionnaires and objective tracking and ECG data were gathered for the statistical analysis.

On the experimental side, we think we can withdraw some initial conclusions, despite of having to finish the statistical analysis for the questionnaires, ECG and head tracking data.

Our preliminary results suggest that subjects could have felt some degree of ownership with the body in front (see Q3b in appendix A) but they did not feel their location changed.
Conclusions

significantly during the experiment, as reported in Q4 and Q7 (see appendix A). Thus we do not think we were able to reproduce the results from [8].

We do not think we achieved a significant degree of disembodiment or change in the visual perspective of subjects. Therefore, we think we did not elicit in any case the illusion of having an out-of-the-body experience.

Subjects feel stronger ownership with the body in front in the Lenggenhager paradigm than in the cord one. That could be because in the cord paradigm there is a collocated body that competes in terms of ownership with the body in front. Subjects may accept their collocated body but they do not easily accept the body change we suggest by means of the tapping during the last part of the experiment. In the Lenggenhager experiment there is only one body in front of subjects. We say that the two bodies clearly compete for the ownership since subjects reported in Q5 (see appendix A) a stronger feeling of having more than one body in the cord paradigm than in the Lenggenhager one.

Visuo-motor contingencies and first person perspective seem to be a dominant combination of factors when it comes to having ownership with a virtual body collocated in a position matching your real physical position. This seems to overwrite the tactile stimulation we provided to subjects since they reported having stronger ownership with the collocated body in the asynchronous condition than in the synchronous one. When there is no collocated body (Lenggenhager paradigm) subjects report more ownership with the body in front in the synchronous condition as expected. In this case subjects do not see their virtual body in first person perspective and only experience visuo-motor contingencies with the body in front.

Our study also shows that subjects can feel a virtual extension coming out of their body as part of their real body when visuo-tactile stimulation is applied synchronously on their body and the virtual extension. In a scale from 1 to 7, they reported an average of 4.1 in the synchronous condition and 2.9 in the asynchronous one.
Potential Applications

10. Potential Applications
The main application of this study is to contribute to the scientific method adding some more knowledge on body perception to the neuroscience and psychology fields by eventually becoming a publication. By better understanding how our brain works and understands our own nature, we will be able to deploy better therapies, HID s and other body related applications. The most immediate long-term applications would be to treat body schema or body ownership disorders like somatoparaphrenia, asomatognosia and others [29] [30].

Although in general there are no direct commercial applications for this study, we think it can spin-off to several other VR projects that better focus on some specific aspects of it.

For instance, these other projects could study ownership on extra limbs or non-human bodily extensions. They could also study OBEs using different ways to attempt to elicit the sense of disembodiment or the sense of embodiment in a non-collocated body.

The applications of the software itself have immediate impact for other VR projects. The automatic tapping me be the preferred method for a number of reasons in some experiments.

The tweener makes it very simple and easy to create complex animations for cameras, objects, lights and so on. Adding this kind of features to a project should no longer mean that we have to spend several hours coding non-reusable and project-specific animations.

For instance, one application of the tweener is that we can easily animate an avatar to look at a specific target and then have the avatar gazing at other targets with smooth transitions in between. To do that we only need to make the avatar look at a dummy (transparent) object that will be animated with tweens to move from one target to the next.
11. Future Work

Our study showed that it is possible to elicit some degree of ownership with virtual bodily extensions by using visuo-tactile remapping. Some other studies could test on other kinds of bodily extensions to further study our brain’s body perception regarding the acceptance of new limbs with different shapes and relative positions.

Another study focusing on the cord could be performed to see under which conditions it is possible to extend the ownership feeling along the cord. In fact, a real time measurement of “ownership length” along the cord was implemented during this study but never used to avoid an overly complex setup. It remains to be seen if we would get the same results for different cord lengths and thicknesses.

Also in the body perception field, some new studies are being carried out using different techniques to try to artificially elicit some kind of out of body experience or disembodiment.

Regarding the software developed, the tweezer classes should be revamped to manage tweens in the same way an operating system manages its processes. Every tween is currently being stored in one single queue. By using multiple queues to manage tweens according to their state the performance would increase and the code would be more scalable and robust. Also, we believe that the next logic step for the tweezer is to become a more complex animation system featuring keyframe based animation. This way the tweezer could handle more than 2 keyframes per transformation with the ability to animate values through complex paths.

The tapping system could also be greatly improved. In the first place, it could be more interactive so that it is much easier and faster to move or delete tapping points without having to delete the whole path every time. In the second place, tapping steps coordinates are currently stored in absolute world coordinates. That could be improved so that the tapping path is relative to the body parts being tapped. For instance, if we want to do tapping on one avatar’s hand and the hand moves, we may want the tapping object to follow the hand and continue the tapping. This way there could be visuo-motor contingencies combined with tactile stimuli on moving parts. That could lead to an increase of the PI since tapping events would react to our movements in VR.

Another improvement that could be added to this project is reactive haptics. In our current implementation, when subjects move the upper part of their body they trigger some movements on the cord. The cord wobbles as a result of those movements but subjects do not
feel any haptics in return. In the same way that we feel a force on our shoulders when somebody pulls one of our hands, it could increase the Psi to provide some haptics in the form of vibrations around the belly (and maybe also on their back) according to the movements of the cord.
12. Plan – Gantt diagram

Figure 53. Estimated Gantt diagram for the project
13. Economic cost

There are several parts we need to take into account to estimate the total cost of this project. Here is a list of the main expenses involved in our study:

- Hardware equipment (InterSense, OptiTrack, HMD)
- Project development (XVR project)
- Subjects for experiments (41 subjects)

We think that a good estimation of the real cost of the project should not include the cost of the hardware that was not bought specifically for this project. We do not include either the cost of those developments that were already done at the time of starting this project or the ones that were not specifically implemented for it.

The total estimated cost of the project is the one detailed in the Gantt diagram in chapter 12 plus the cost of paying 41 subjects (each subject was payed 10€). Most of the documentation was developed outside work hours, thus it is not added to the final cost.

The next table details the estimated overall cost of the project.

<table>
<thead>
<tr>
<th>Item</th>
<th>Price (€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preliminary design</td>
<td>874,50</td>
</tr>
<tr>
<td>3D design</td>
<td>1221,00</td>
</tr>
<tr>
<td>Software development</td>
<td>2992,00</td>
</tr>
<tr>
<td>Pilots</td>
<td>616,00</td>
</tr>
<tr>
<td>Experiments</td>
<td>968,00</td>
</tr>
<tr>
<td>Subjects Payment</td>
<td>410,00</td>
</tr>
<tr>
<td>Data Analysis</td>
<td>1232,00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>8313,50€</strong></td>
</tr>
</tbody>
</table>

Figure 54. Quotation for the project
14. References


References


15. Appendix A. Questionnaires and Other Experiment Documents

15.1. Questions for the no cord condition (Lenggenhager 2007 paradigm)

Q1 It seemed as if I were feeling the touch of the ball in the location where I saw the virtual body touched.
1 2 3 4 5

Q2 It seemed as though the touch I felt was caused by the ball touching the virtual body.
1 2 3 4 5

Q3 It felt as if the virtual body was my body.
1 2 3 4 5

Q4 It felt as if my (real) body was drifting towards the front (towards the virtual body).
1 2 3 4 5

Q5 It seemed as if I might have had more than one body.
1 2 3 4 5

Q6 It seemed as if the touch I was feeling came from somewhere between my own body and the virtual body.
1 2 3 4 5

Q7 It appeared (visually) as if the virtual body was drifting backwards (towards the real body).
1 2 3 4 5

Q8 During the experiment the body I saw was that of another person
1 2 3 4 5

Q9 When the fan descended over the head of the virtual body in front, it felt as if it could chop my head
1 2 3 4 5
15.2. Questions for the cord condition

Q1 It seemed as if I were feeling the touch of the ball in the location where I saw the virtual body touched.

   c) The body in my own position
   1 2 3 4 5

   d) The body in front
   1 2 3 4 5

Q2 It seemed as though the touch I felt was caused by the ball touching the virtual body.

   c) The body in my own position
   1 2 3 4 5

   d) The body in front
   1 2 3 4 5

Q3 It felt as if the virtual body was my body.

   c) The body in my own position
   1 2 3 4 5

   d) The body in front
   1 2 3 4 5

Q4 It felt as if my (real) body was drifting towards the front (towards the virtual body).

   1 2 3 4 5

Q5 It seemed as if I might have had more than one body.

   1 2 3 4 5

Q6 It seemed as if the touch I was feeling came from somewhere between my own body and the virtual body.

   1 2 3 4 5

Q7 It appeared (visually) as if the virtual body was drifting backwards (towards the real body).
1 2 3 4 5

Q8 During the experiment the body I saw was that of another person
   c) The body in my own position

1 2 3 4 5

   d) The body in front

1 2 3 4 5

Q9 When the fan descended over the head of the virtual body in front, it felt as if it could chop my head

1 2 3 4 5

Q10 When the ball touched the cord, did you feel that the touch was coming from where the ball touched the cord? If yes, indicate how far along the cord this happened.

15.3. Experiment Information for Subjects

This experiment is part of a series where we are attempting to learn about people’s responses to virtual reality experiences. In this experiment you will wear a head-mounted display that
will show you a virtual world. You will be standing. You will also be wearing a haptic vest with built in vibrators and a pair of globes with trackers.

In this experiment you will be transported to a virtual concrete room with several objects around you. It will be a very quiet environment with some music.

You will be asked to explore the room and move your body and notice what changes as a result in the virtual reality. Afterwards, we will ask you to pay attention to some events happening in the virtual world for several minutes. The whole experiment will take about half an hour. We will explain you the purposes of the experiment after finishing it. We will pay you 10€ for your participation.

If you have any questions, now it would be a great moment to let us know! And remember that you are free to give pull out of the experiment at any time without giving explanations.

It is important to note that there may be events that apparently occur in the virtual environment that some people might regard as stressful. If you have any doubts about this please discuss with the experimenter or reconsider whether you wish to take part in the study.

15.4. **VR Experience**

The following questions will be read out to the participants at the end of each experience.

How much do you agree with the following statements about your experience? Each statement should be answered with:
Appendix A. Questionnaires and Other Experiment Documents

1. Disagree strongly
2. Disagree somewhat
3. Neither agree nor disagree
4. Agree somewhat
5. Strongly agree

General questions

1. I felt as if the body I saw in front of me might be my body.
2. I had the feeling that I might be harmed when the fan descended over the avatar in front of me.
3. The movements I saw the body in front was making seemed to be my movements.
4. The body I saw in front was another person.
5. What do you think about the experience? How did you feel during the experiment?

Additional questions for the cord condition

6. I felt as if the cord was an extension of my body.
7. I felt as if I had two bodies.
8. I felt as if I was located at some point in the space in between the two virtual bodies.

15.5. Order of Procedures for the Experiment

Pre experiment steps

1. Volunteer is recruited and at the time of recruitment is given the trauma questionnaire and the pre-questionnaire.
2. They must ‘pass’ the trauma questionnaire to be allowed to do the experiment.
3. Person arrives to the experiment they are assigned an id number
4. They are given the information sheet
5. They are asked if they wish to continue
6. If yes they are given the consent form to sign
7. The relaxation/stress scale is explained
8. The experiment is explained to them verbally

Setup
Appendix A. Questionnaires and Other Experiment Documents

Setup subject:

9. Put ECG electrodes and other physio equipment
10. Setup vibrators vest
11. Setup hand tracking

On the tracking computer (PCB):

12. Run tracking servers (intersense & optitrack). vrpn_server, .bat, tracking tools
13. Run nexus physio server

On the experiment computer (PCC):

14. Run XVR project windowed
15. Make sure XVR connects to the correct COM port (should be COM 4 in the experiments PC in the lab)
16. (You will see on the output the message "First Connected COM device on Port: <#port>")
17. Run XVR project in full screen
18. Run triple client to make recorded data persistent in a text file

Testing stage

19. Run the XVR project
20. (p) test vibrators
21. Put HMD on subject
22. Test sound on headphones
23. Put headphones on the subject

Experiment

24. Update iPhone slate with id number
25. Video ON. Start the video and put slate in ACTION mode
26. (wand 2 / mouse mid button) Calibrate HMD position in virtual environment
27. (wand 8 / a) Calibrate arms tracking for the IK to work correctly
28. Switch on the HMD
29. Ask subjects to look around and play with IK on upper body parts (move head, move upper body part, move arms, look at the cord)
30. (t+a) Toggle ASYNC tapping mode on if necessary for this condition
31. (s) Start tapping. Pay attention to what you feel on your body while the tapping is running
32. Experimenter waits until fan descends
33. At the end switch OFF the video and put the iPhone slate in CUT mode
34. Switch off the HMD

Post experiment steps
35. Interview the subject right after turning of the HMD
36. Then they complete the two post-experiment questionnaires, first the shorter one, then the longer one
37. Start the video again
38. Do the interview that explores their reactions in more detail
39. Switch off video
40. They are paid (not applicable for pilots), and asked to sign the payment form, and told not to talk about it with anyone for 2 months

SHORTCUTS

(t+a) to toggle async tapping mode on

Other functions:

GENERAL MENU

(c) toggle cord on/off

(m) toggle ruler

(s) unbalance avatar with sudden motion

(l) throw last ball

(g) to change ground

(y) trigger sudden motion

(1) toggle 1/2 tapping balls

TAPPING MENU

(q) quit from tapping menu

(a) toggle async tapping mode on
### 16. Appendix B. Questionnaire Results

#### 16.1. Lenggenhager paradigm

<table>
<thead>
<tr>
<th>Subject ID</th>
<th>Condition</th>
<th>Q1b</th>
<th>Q2b</th>
<th>Q3b</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
<th>Q7</th>
<th>Q8b</th>
<th>Q9</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>The fan was a surprise</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>29</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>34</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>38</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

**Average**

|   | 1   | 4,7 | 4,1 | 3,6 | 2,4 | 2,2 | 2,3 | 2,1 | 2,4 | 2,8 |

| 4 | 2   | 4   | 4   | 3   | 4   | 4   | 4   | 4   | 2   | 2   | I felt surprise |
| 6 | 2   | 3   | 5   | 3   | 5   | 5   | 1   | 1   | 1   | 4   | When I saw the fan my first reaction was to kneel on my knees. |
| 11 | 2  | 1   | 4   | 1   | 1   | 1   | 3   | 2   | 5   | 3   | It was like controlling a marionette. Perspective not perfect. I saw the fan already down. |
| 16 | 2  | 1   | 1   | 2   | 3   | 1   | 1   | 1   | 4   | 3   | Really immersive. Ball did not match very well the vibrations. First reaction was to avoid the fan. |
| 19 | 2  | 1   | 4   | 1   | 2   | 1   | 2   | 2   | 1   | 3   | Fan scared me because it was close to my point of view. |
| 24 | 2  | 1   | 3   | 2   | 1   | 3   | 1   | 1   | 2   | 2   | The avatar in front was my projection. My first reaction with the fan was to get out of its way, like in videogames. |
| 27 | 2  | 1   | 1   | 2   | 3   | 1   | 4   | 1   | 1   | 2   | Little link with the avatar in front. |
| 32 | 2  | 2   | 2   | 3   | 2   | 1   | 1   | 2   | 4   | 2   | Arms movement very fake. It broke the illusion. |
| 36 | 2  | 2   | 2   | 3   | 2   | 2   | 4   | 2   | 3   | 3   | The fan was a bit stressing. |
| 37 | 2  | 2   | 2   | 3   | 2   | 2   | 4   | 2   | 3   | 3   | |

**Average**

|   | 2   | 2   | 2,8 | 2,5 | 2,6 | 2,1 | 2,6 | 1,7 | 3   | 2,5 |
## 16.2. Virtual cord paradigm

<table>
<thead>
<tr>
<th>Subject ID</th>
<th>Condition</th>
<th>Q1 a</th>
<th>Q1 b</th>
<th>Q2 a</th>
<th>Q2 b</th>
<th>Q3 a</th>
<th>Q3 b</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
<th>Q7</th>
<th>Q8 a</th>
<th>Q8 b</th>
<th>Q9</th>
<th>Q10</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>Immersive, creepy, interesting. Cord felt very real when tapping started on the cord.</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>The cord made me nervous, anxious. It was really unpleasant. I wanted to stop the experience.</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>I needed some time to get used to the environment. No intention to avoid the fun.</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>My body feels very real. Cord tapping very real when the ball is close to me.</td>
</tr>
<tr>
<td>23</td>
<td></td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>I thought I would feel something when the fan was chopping the front avatar's head.</td>
</tr>
<tr>
<td>31</td>
<td></td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>The avatar in front was mimicking me. I felt the cord had weight.</td>
</tr>
<tr>
<td>33</td>
<td></td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>I was wondering when would the fan stop.</td>
</tr>
<tr>
<td>40</td>
<td></td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Impressed to feel on my body the tapping on the cord. No worried about the fan. I knew it was a virtual fan.</td>
</tr>
</tbody>
</table>

| Average    |           | 3    | 4.7  | 4.1  | 4.4  | 4.7  | 3.8  | 2.7 | 1.6 | 2.4 | 2.6 | 1.7  | 1.7  | 3    | 2    | 1    | 4    |

| 2          |           | 4    | 1    | 1    | 3    | 1    | 4    | 5  | 2   | 4   | 2   | 2    | 1    | 2   | 5    | I felt dizzy. |
| 8          |           | 4    | 4    | 2    | 4    | 2    | 4    | 4  | 2   | 5   | 3   | 2    | 1    | 1   | 5    | First thought was to avoid the fan, it is instinctive. |
| 12         |           | 4    | 2    | 2    | 4    | 5    | 5    | 4  | 1   | 5   | 4   | 1    | 1    | 1   | 4    | The fan was stressful. I already knew about astral travels. |
| 15         |           | 4    | 4    | 2    | 4    | 2    | 4    | 3  | 2   | 2   | 4   | 2    | 1    | 2   | 5    | The fun is about to chop the head of the avatar in front. |
| 17         |           | 4    | 2    | 3    | 2    | 2    | 4    | 3  | 1   | 4   | 1   | 1    | 1    | 1   | 2    | Ball did not match vibrations. Not afraid of the fan. |
| 21         |           | 4    | 4    | 2    | 2    | 2    | 5    | 4  | 2   | 1   | 4   | 1    | 4    | 2   | 2    | |
| 25         |           | 4    | 1    | 1    | 2    | 2    | 5    | 3  | 2   | 1   | 1   | 1    | 1    | 2   | 4    | |
| 30         |           | 4    | 4    | 2    | 4    | 1    | 5    | 1  | 1   | 1   | 2   | 1    | 1    | 3   | 4    | Like in a videogame with 1st and 3rd person at the same time. I was expecting. |
Appendix B. Questionnaire Results

<p>| | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>39</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

- some tactile feedback from the fan.
- No consequences on my person coming from the fan.
- Stressing experience at the beginning. I doubted about what to do when I saw the fan coming down.

<table>
<thead>
<tr>
<th>Average</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2,7</td>
<td>2,3</td>
<td>2,8</td>
<td>2,3</td>
<td>4,5</td>
<td>3,2</td>
<td>1,8</td>
<td>3</td>
<td>2,6</td>
<td>1,6</td>
<td>1,7</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3,7</td>
<td>2,9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>