



Department of Informatics
Technical University of Munich



TECHNICAL UNIVERSITY OF MUNICH

DEPARTMENT OF INFORMATICS

MASTER'S THESIS IN INFORMATICS: GAMES ENGINEERING

**Combination of Myo and HTC Vive for Navigating in the
Neurorobotics Platform**

Tamara Barounig

TECHNICAL UNIVERSITY OF MUNICH
DEPARTMENT OF INFORMATICS

Master's Thesis in Informatics: Games Engineering

**Combination of Myo and HTC Vive for
Navigating in the Neurorobotics Platform**

**Verbindung von Myo und HTC Vive für die
Navigation in der Neurorobotics Platform**

Author:	Tamara Barounig
Supervisor:	Prof. Gudrun Klinker, Ph.D.
Advisor:	Sandro Weber, M.Sc.
Date:	April 20, 2018

I confirm that this Master's Thesis is my own work and I have documented all sources and material used.

Garching, April 20, 2018

Location, Date

Signature

Acknowledgement

After my thesis is finished, it is my very pleasure to thank all the people who helped and supported me during my work.

First of all, I want to express my gratitude to Prof. Gudrun Klinker, Ph.D., who gave me the opportunity to write my thesis at her chair.

This work would not have been possible without Sandro Weber, M. Sc. who introduced me to the Myo, the HTC Vive and the Neurorobotics Platform (NRP) and offered to be my advisor. I would like to thank him for our weekly meetings and his help.

Special thanks go to all the people who willingly shared their precious time by participating in my user studies. Your feedback and ideas regarding the navigation methods were really great and helped me a lot. You really saved my thesis.

I want to express my utmost thanks to my boyfriend Felix for always being there for me, for supporting me, for offering ideas and for reading over my entire thesis. You do not know how much all of this means to me. This whole project would not have been manageable without you by my side.

Last but not least I want to thank my family and especially my parents for their unconditional love and care. It is not always easy with you, but I love all of you from the bottom of my heart and I would not have come this far without you. My little sister Sonja has been my best friend all my life and I love her dearly and thank her for all her advice and support. You are most likely the only person in the world who seems to fully understand me and my quirks. Thanks for always listening to me.

Abstract

Virtual Reality (VR) is advancing quickly and recently even found its way into the private households. As it provides many potentials and allows an immersive view into the virtual environment it is also used in different research areas. One of these areas includes the research on the human brain, which is fostered in the scope of the Human Brain Project (HBP) and the Neurorobotics Platform (NRP).

Although VR has come quite far in the last couple of years, we still lack an intuitive and immersive way to navigate in virtual environments. Thus, this thesis proposes a novel and intuitive gesture-based navigation approach which utilizes the Myo and the HTC Vive. Therefore, we first designed different gesture-based navigation methods which were tested against each other to find out which method is most suitable to be revised and implemented within the scope of the NRP. Finally, a refined method was developed and compared against a controller-based one to evaluate its potential and flaws.

CONTENTS

1	Introduction	1
1.1	Motivation	1
1.2	Main Contributions	2
1.3	Outline	3
2	Background	5
2.1	The Human Brain Project (HBP) and the Neurorobotics Platform (NRP)	5
2.1.1	Structure of the Neurorobotics Platform	7
2.1.2	Robot Operating System (ROS)	8
2.1.3	Gazebo	10
2.1.4	Unity3D-Client	12
2.2	Virtual Reality (VR)	13
2.2.1	The HTC Vive	15
2.3	Gesture-Based Input	15
2.3.1	The Myo	16
3	Related Work	19
3.1	Gesture-Based Avatar Navigation in VR	19
3.2	Recent Works on the Myo	23
3.2.1	Myo in VR	23
3.2.2	Myo for Navigation	23
3.2.3	Myo for Navigation in VR	24
4	Comparison of Different Gesture-Based Navigation Approaches	27
4.1	Approach	27
4.1.1	Controlling the Begin and End of a Motion	28
4.1.2	Controlling the Movement's Speed	30
4.1.3	Controlling the Movement's Direction and User's Rotation	30
4.1.4	Resulting Navigation Methods	30

4.1.5	Vibration Feedback	31
4.2	Implementation	31
4.2.1	Utilized Tools and Technologies	32
4.2.2	Implementation of the Avatar	33
4.2.3	Implementation of the Navigation Methods	35
4.3	Evaluation	37
4.3.1	Goals	37
4.3.2	Preselection	37
4.3.3	Setup and Structure	37
4.3.4	Practical Part	40
4.3.5	Questionnaire	42
4.4	Results	43
4.4.1	Demographic Profile	43
4.4.2	Subjective Assessment of the Navigation Methods	43
4.4.3	Objective Assessment of the Navigation Methods	49
4.4.4	Aggregated Feedback	53
4.4.5	Summary and Implications	54
5	Comparison of Gesture- and Controller-Based Navigation	57
5.1	Approach	57
5.1.1	Refinement of the Gesture-Based Navigation Method	58
5.1.2	Design of the Controller-Based Navigation Method	59
5.2	Implementation	60
5.2.1	Utilized Tools and Technologies	60
5.2.2	Changes on the Unity3D-Client	60
5.2.3	Implementation of the Physical Avatar-Rotation	62
5.2.4	Implementation of the Gesture-Based Navigation Method	63
5.2.5	Implementation of the Controller-Based Navigation Method	64
5.3	Evaluation	64
5.3.1	Goals	64
5.3.2	Setup and Structure	65
5.3.3	Practical Part	67
5.3.4	Questionnaire	68
5.4	Results	71
5.4.1	Demographic Profile	71
5.4.2	Subjective Assessment of the Navigation Methods	72
5.4.3	Objective Assessment of the Navigation Methods	77
5.4.4	Aggregated Feedback	78

5.4.5	Summary and Implications	80
6	Discussion and Conclusion	83
6.1	Conclusion	83
6.2	Future Work	85
A	Appendix	89
A.1	Unity3D-Client Sequence Diagram	89
A.2	User Study: Different Navigation Approaches	93
A.2.1	Figures Used to Explain the Different Navigation Approaches . .	93
A.2.2	Information for the Users	95
A.2.3	Questionnaire	97
A.2.4	Data Gathered During the Practical Part	102
A.2.5	Detailed Survey Results	106
A.3	User Study: Comparison of Gesture- and Controller-based Navigation .	109
A.3.1	Figures Used to Explain the Different Navigation Approaches . .	109
A.3.2	Information for the Users	110
A.3.3	Questionnaire	112
A.3.4	Data Gathered During the Practical Part	115
A.3.5	Detailed Survey Results	119
B	List of acronyms	121
	Bibliography	122

LIST OF FIGURES

2.1	Logo of the Human Brain Project from [Eurnda].	5
2.2	Overview of the NRP's components, with the rows representing the three feature-types design and editing (Design), simulation (Run) and interactive visualization (Visualize). [Fal17a]	7
2.3	An overview of the basic communication concept used in ROS. [Mar14b]	9
2.4	Overview of the Myo with its sensors and expandable casing from [Thandd].	16
2.5	Myo gestures from left to right: Wave left, wave right, double tap, fist, fingers spread. Taken from [Thandf].	17
3.1	The left picture show the ten motions supported by the system and the right picture shows the corresponding hand postures needed to trigger these motions.	20
3.2	The left picture shows the eleven motions supported by the system and the right picture shows the corresponding hand gestures needed to trigger these motions.	20
3.3	Overview of the gestures used in the method proposed by Codd-Downey et al. [Cod14]. From left to right: forward, reverse, turn and stop. . . .	22
3.4	Gesture mapping of movements and turns to the left and right hand. . .	22
4.1	Break down of the Myo's orientation into yaw, roll, and pitch for an easier understanding. [Mar14a].	28
4.2	Illustration of the "ArmPitch" method, showing that raising the arm leads to a forward movement and lowering the arm to a backward movement.	29
4.3	Unity Hierarchy-view of the GameObject Player.	34
4.4	The Myo's coordinate system in Unity.	36
4.5	A schematic overview of the evaluation procedure.	38
4.6	View onto the test area within Unity.	40

4.7	Top-down view on the parkour with the five most important exercises zoomed-in and highlighted in red. The white circle in the middle of the parkour represents the user and indicates the starting position. The highlighted elements are: (1) Pallet working as a switch to move the shelf. (2) Shelf. (3) Barrel-slalom. (4) Narrow passage with stacked crates. (5) Slalom made of bottle-cases.	41
4.8	Bar chart showing the users' difficulties with the methods.	44
4.9	Bar chart depicting the users' types of difficulty.	44
4.10	Bar charts visualizing the survey results regarding motion sickness. The blue bars show how many of the 29 participants experienced motion sickness and the gray bars show the mean sickness score of these people on a scale from 1 (only a little dizzy) to 5 (very dizzy).	46
4.11	Bar charts visualizing the survey results regarding pain. The blue bars show how many of the 29 participants felt any pain and the gray bars show the mean pain score of these people on a scale from 1 (only a little) to 5 (very much).	46
4.12	Point-based ranking of the different navigation methods created from the users' rankings in the questionnaire.	48
4.13	Visualization of two statistical measures representing the time needed to complete the parkour with the different navigation methods.	49
4.14	Visualizations showing the number of different objects hit by all users and how often users hit objects.	50
4.15	Visualization of the users' collisions with the different navigation methods. The color indicates how many different users collided with an object while using the specified navigation method.	52
5.1	Visualizations showing how the refined gesture-based navigation method works.	58
5.2	Visualization of the controller-based navigation method.	59
5.3	A schematic overview of the evaluation procedure.	65
5.4	View onto the test area within the NRP.	67
5.5	Top-down view on the parkour with the five most important exercises zoomed-in and highlighted in red. The user starts in front of the glass door on the top right. The highlighted elements are: (1) Slalom made of office chairs. (2) Passage created with the wall and the padded chair. (3) Smaller slalom made of dustbins. (4) A narrow passage between wall and couch. (5) Pioneer robot as target.	69

5.6	Top-down view on the balance test level with the five most important exercises zoomed-in and highlighted in red. The user starts on the bottom right of the level. The highlighted elements are: (1) View on the glass cage trapping the robot. (2) Stairs which the user needs to use to get onto the small runway. (3) View onto the small runway which the user needs to balance on to get to the robot. (4) A gap in the glass cage which allows the user to get down to the robot. (5) Husky robot as target. . .	70
5.7	Pie chart showing the users' age.	72
5.8	Bar chart depicting the users' assessment of the type of difficulty.	73
5.9	Bar chart depicting the users' assessment of the method's precision on a scale between 1 (very imprecise) and 5 (very precise).	74
5.10	Bar chart depicting how the users felt while using the method on a scale from 1 (very bad) to 5 (very good).	74
5.11	Bar chart visualizing the survey results regarding motion sickness on a scale from 1 (only a little dizzy) to 5 (very dizzy).	75
5.12	Visualization of the users' preferred navigation method.	76
5.13	Visualization of two statistical measures representing the time needed to complete the parkour and the balance level.	77
5.14	Illustration of the mean number of drops registered during the balance level.	78
A.1	Explanation for the method: Hold fist to move and gaze direction for steering	93
A.2	Explanation for the method: Hold fist to move and hand direction for steering	93
A.3	Explanation for the method: Use start and stop gestures to move and gaze direction for steering	94
A.4	Explanation for the method: Use start and stop gestures to move and gaze direction for steering	94
A.5	Explanation for the method: Use the pitch of the arm to move and gaze direction for steering	94
A.6	Explanation for the method: Use the pitch of the arm to move and gaze direction for steering	94
A.7	Statistics regarding the users' experience with VR headsets.	106
A.8	Statistics outlining the users' knowledge of the Myo and their experience with it.	106
A.9	Bar charts showing the users' placements of the different navigation methods.	107

A.10 This chart shows the mean rating the users assigned to the different navigation methods in terms of their feeling while utilizing this method. Thereby, 1 indicates a very bad feeling and 5 a very good one.	108
A.11 Explanation for the controller-based navigation method	109
A.12 Explanation for the gesture-based navigation method	109
A.13 Statistics regarding the users' experience with VR headsets.	119
A.14 Statistics outlining the users' knowledge of the Myo and their experience with it.	119

LIST OF TABLES

3.1	Table from McCullough et al. showing the mean responses of the users on a 5-point Likert scale, where 5 indicates high agreeableness. [McC15]	25
4.1	Utilized tools and technologies with the appropriate versions and specifications.	32
5.1	Utilized tools and technologies with the appropriate versions and informations.	61

CHAPTER 1

INTRODUCTION

This chapter presents the incentive behind this thesis as well as its main contributions and gives a short overview of the following chapters.

1.1 MOTIVATION

From the moment we are born our brain starts working, and doesn't stop until the second we die. It is one of the most powerful and complex objects on this world, and although we use it every second, we are still not able to fully understand how it works. The European Commission recognized how challenging and important this topic is and initiated the Human Brain Project (HBP). The aim of this project is to foster the cooperation and communication between scientists of all fields to advance the research for understanding the human brain.

One part of the HBP is the Neurorobotics Platform (NRP), which allows scientists to apply brain models to robots and analyze them in sensory-rich environments. This does not only provide new opportunities for neuroscientists, but also for roboticists which can use this platform to invent new kinds of artificially intelligent robots. Such robots may be used in the real world to support elderly people or help in manufacturing. Hence, their brain models should be able to react to and interact well with humans. Thus, the human-robot interaction need to be analyzed beforehand. Therefore, the simulations provided by the NRP can be used.

However, the analysis of human-robot interaction is quite difficult, due to the fact that we can't simulate humans or human behavior. Therefore, behavior of real humans has to be projected into the simulation through the use of Virtual Reality (VR). Since, the

human behavior is very complex and consists of a variety of aspects, this thesis only focuses on projecting human movements into the virtual environment.

To ensure that the human-robot interaction is realistic, the human needs to be able to move freely in the virtual space. This not only includes going forward, backward, left and right but all kinds of movements and interactions which are possible in the real world. As this thesis is not able to cover every kind of human motion, the focus is placed on basic human movements.

This leads to a frequently addressed question in VR-research: What is the best method to map human motions from the real world into the virtual one, without influencing or limiting the movements through the used technique?

Exploring all possibilities would exceed the scope of this thesis by far. Hence, this thesis focuses on developing and evaluating new interaction techniques by using the Myo armband, which is described further in section 2.3.1.

1.2 MAIN CONTRIBUTIONS

The main contribution of this thesis is the exploration of possible gesture-based interaction techniques provided by the use of the Myo armband together with the HTC Vive. The corresponding research question is: Is the combination of Myo and HTC Vive capable of mapping human motions into the virtual space without feeling unnatural and disconnected from the virtual environment?

During the first part of this thesis different gesture-based navigation methods were designed and implemented in a basic Unity project. Subsequently, these methods were evaluated in a preliminary user study which pursued two goals. The first goal was to find out which method is most suitable for navigation in VR and the second goal was to gather as much user feedback as possible to refine the concepts and adapt them to the user's expectations and needs.

After the assessment of the preliminary user study, the favored method was re-implemented in NRP's Unity3D-Client. The client is used as a base project for the second part of this thesis and in charge of the connection to the NRP. In addition to the improved concept, there was also a controller-based navigation approach implemented in the client to allow a comparison between gesture- and controller-based navigation.

At the end, a final user study was conducted which compared the two approaches

against each other to evaluate if the proposed navigation method offers advantages over established controller-based methods.

1.3 OUTLINE

This section briefly describes the outline of the present thesis.

At first the thesis' preliminary setting is described. Therefore, a short explanation of the thesis's background and some fundamental concepts and tools are introduced in **chapter 2**. There, information on the HBP and the NRP as well as on VR and gesture-based input is presented. The following **chapter 3** contains a brief overview of already published work on gesture-based avatar navigation in VR and research focusing on the potentials of the Myo in VR as well as for navigation.

Chapter 4 presents the development and evaluation of different, gesture-based navigation methods. It provides an overview of the methods' design and their implementation in Unity. Thereby, different design possibilities are outlined and compared with each other. The evaluation method is presented on the basis of its goals, setup and structure, whereas, the designs of the practical part and the questionnaire are explained in detail.

The revisioning of the favored navigation approach and its implementation in the NRP's Unity3D-client is discussed in **chapter 5**. The chapter also contains the implementation of a controller-based navigation method into the client. It also comprises the final user study which compared the controller-based navigation approach with the gesture-based one. Therefore, the evaluation method, the structure of the user study and the corresponding results are explained and analyzed.

The whole thesis is summed up and reviewed in **chapter 6**, which also includes a discussion of potential future developments and improvements.

Appendix A includes a conceptual overview of the Unity3D-client as well as the explanations and questionnaires used for the two user studies. It also includes additional survey results which are not discussed in the main part of this thesis.

CHAPTER 2

BACKGROUND

This chapter describes the setting in which this thesis was written. Therefore, the chapter starts with an explanation of the HBP and the NRP. It continues with the description of two already implemented components used in this thesis. Then the basic concepts of VR and gesture-based input are explained with a focus on the respective technologies used in this thesis.

2.1 THE HUMAN BRAIN PROJECT (HBP) AND THE NEUROROBOTICS PLATFORM (NRP)

The HBP is a flagship project of the HORIZON 2020, the EU Framework Program for Research and Innovation which started in October 2013. The goal of this program is to foster long-term European collaborative research, address major European challenges and create new opportunities and innovations. [Eurndb]



Human Brain Project

Unifying our understanding of the human brain.

FIGURE 2.1: Logo of the Human Brain Project from [Eurnda].

In this context, the HBP is dealing with one of the greatest challenges scientists are facing in the 21st century: The comprehension of the human brain. This not only includes the functioning and structure of the brain, but also its computational capabilities. Therefore, the main goal of the project is to create a computer simulation of the human brain.

To allow for an easier analysis and emulation, the platform provides mathematical models and Information and Communications Technology (ICT) tools to researchers worldwide. Thus, scientists can use this platform to collaborate with each other across disciplines. [Eurndb; Humnda; R 16]

In the course of this project the following six ICT-based research Platforms were created:

1. Neuroinformatics Platform
Create an atlas to explore the brain.
2. Brain Simulation Platform
Digital brain reconstruction and simulation.
3. High Performance Analytics and Computing Platform
Provide software and storage for the whole project.
4. Medical Informatics Platform
Allow medical scientists to share medical data and generate disease models.
5. Neuromorphic Computing Platform
Brain-like computation.
6. Neurorobotics Platform
Control robots through a brain model.

[Humnda; Humndb]

The NRP is the sub-project in which frame this thesis was written. It allows scientists to combine brain models with a variety of different robots and analyze their behavior in complex virtual environments. Therefore, the brain models are simulated based on spiking neural networks. Neuroscientists are able to validate these brain models on virtual guinea pigs, which would be very difficult or impossible to perform in the real world. In addition to that, roboticists are provided with new possibilities to control their robots.

To use the platform, scientists just need to login and create a simulation with the desired environment, robot and brain model. For collaboration purposes, existing projects can be launched and running simulations can be joined. Thereby, researchers around the world can work together in analyzing the brain model's behavior. [Neund; Hum17b; Fal17b]

2.1 THE HUMAN BRAIN PROJECT (HBP) AND THE NEUROBOTICS PLATFORM (NRP)

2.1.1 STRUCTURE OF THE NEUROBOTICS PLATFORM

The NRP can be roughly divided into three main feature-types which can be summarized as design and editing, simulation and interactive visualization. These features can be used to categorize the different parts of the platform as seen in figure 2.2.

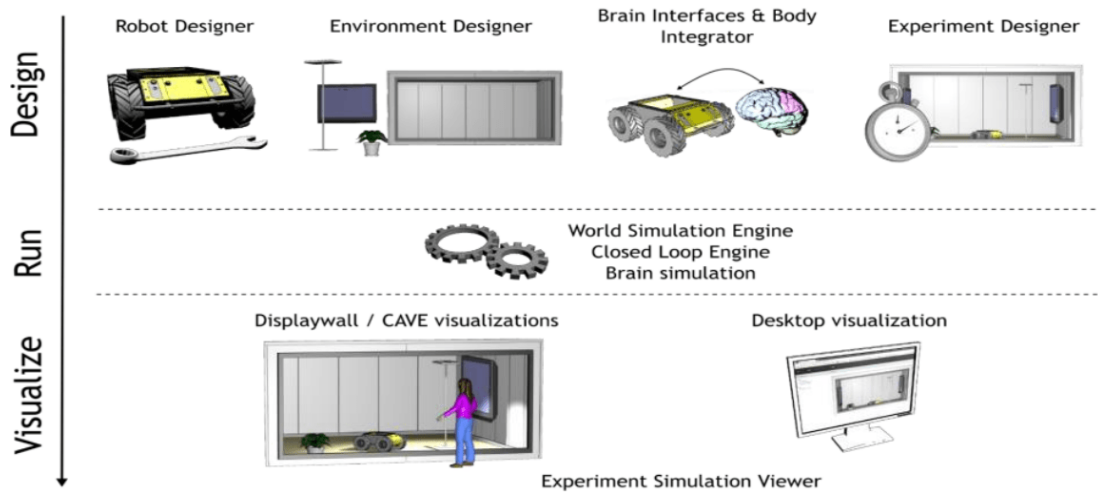


FIGURE 2.2: Overview of the NRP’s components, with the rows representing the three feature-types design and editing (Design), simulation (Run) and interactive visualization (Visualize). [Fal17a]

Design and Editing: Defining all properties of a simulation is very important in conducting good and reproducible experiments. To meet this requirement, the NRP offers different designers and editors. One of them is the **Robot Designer** which is a Blender-Plugin used to define the appearance of the robot as well as its kinematic and dynamic properties. As the robot’s brain responds to the environment, it is very important that researchers are able to use the **Environment Editor** to define how this environment is designed. The **Brain Editor** allows the user to either design a brain model out of scratch or by using an already existing one. To connect the brain with the robot simulations, the **Brain Interface & Body Integrator** is used. This tool translates the sensor data from the robot into suitable input data for the brain and the output data from the brain into movement commands for the robot. For this purpose, the tool uses special transfer functions, which are Python scripts that can be adapted via the **Transfer Functions Editor**. To complete the tool-set the **Experiment Workflow Editor** is provided which uses a state machine to add environment events to the simulation. With this, the platform’s user is also able to interact with some objects in the environment without having to stop the simulation. [Fal17b; Hum17a]

Simulation: After the setup of the experiment, different simulators are used to run the simulation. In order for the brain to send inputs to the robot, it needs to be simulated through the **Brain Simulator**. This tool uses the neuron simulator NEST, which is especially good for spiking neural networks like the ones used to implement the brain models in the NRP. As the brain model is now able to send data to the robot, the robot must also be able to react to this data. Therefore, the **World Simulator** is needed, which uses Gazebo (c.f. section 2.1.3) to provide a physically correct simulation of the robot and the environment. To synchronize the different simulations with each other and guarantee the data transfer between the simulations and the transfer functions, the **Closed Loop Engine** is needed. This software component is the core of the NRP and defines the unique logic of each experiment. In addition, it is also responsible for the smooth interplay between the simulators, designers and the visualization tools. [Fal17b; Thend]

Interactive visualization: The platform's user needs a possibility to control the simulation, watch it, interact with it and take some measurements. Therefore, the **Experiment Simulation Viewer** provides the user with a GUI where he can start, stop and pause simulations, edit the experiment, walk around in the simulation and interact with the robot. Moreover, the viewer allows multiple users to watch the same experiment concurrently, which allows for closer collaboration and analyzation of the experiment. To allow interaction with the simulation, the different components implement their services in the Robot Operating System (ROS) (c.f. section 2.1.2) which allows easy access through ROS topics. On the side of the user interface the interaction is handled via RESTful APIs. [Fal17b]

2.1.2 ROS

ROS is an open-source robotics software platform, which is used in the NRP to implement the different services needed to interact with the simulation. It aims to simplify the generation of complex robot behavior by promoting code reuse and collaboration between developers. Like any other operating system, it provides hardware abstraction, low-level device control, and message passing. In contrast to many other robotics software platforms, ROS does not only provide a variety of different tools but also allows individual design and loose coupling of executables and supports the collaboration of developers by providing a community network. In addition to that, it can be easily implemented in any modern programming language and is able to deal with large runtime systems. However, it only runs on Unix-based platforms at the moment. [Tho14; Rosnda]

2.1 THE HUMAN BRAIN PROJECT (HBP) AND THE NEUROBOTICS PLATFORM (NRP)

Conceptually speaking, ROS is structured into three levels: the Filesystem level, the Computation Graph level and the Community level.

The Filesystem level covers all kinds of packages, message types, and service types. A package thereby forms the smallest possible unit which can be build and released.

To process all the data the different ROS processes (nodes) form a peer-to-peer network which is denoted as the Computation Graph. This network uses different concepts like nodes, master, parameter server, messages, topics, services, and bags to provide data. To ensure that all nodes know each other and are able to exchange messages, the master acts as a nameservice. To route messages, a publish/subscribe system with different topics is used. A topic is an identifier for certain kinds of data. As figure 2.3 shows, a node can publish its data to a specific topic. Any node which is interested in this data can subscribe to the specific topic and then receives newly published data. To provide request/reply interactions as well, a system of services is used as indicated by the dotted arrow in figure 2.3. The last concept used in the Computation Graph are bags, which are used to store and playback messages. [Mar14b]

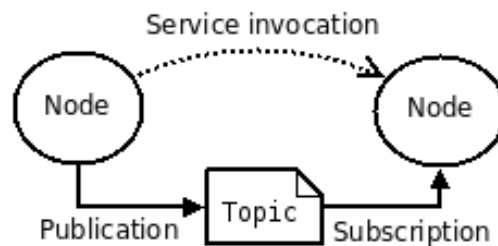


FIGURE 2.3: An overview of the basic communication concept used in ROS. [Mar14b]

The Community level is used to exchange knowledge and software with separate communities. Therefore, resources like distributions, repositories, the Wiki, mailing lists and a bug ticket system are used. [Mar14b]

MESSAGES

A basic data structure used in ROS are messages which support primitive data types like integer, floating point and boolean as well as arrays and nested structures. Each message uses a message type consisting of the package name concatenated with the name of the `.msg`-file through a slash. [Kur16]

To display information on the message types the command-line tool `rosmmsg` is used. According to the command-line documentation of `rosmmsg`, the following commands are currently supported:

CHAPTER 2: BACKGROUND

<code>rosmmsg show</code>	<code>show message description</code>
<code>rosmmsg info</code>	<code>alias for rosmmsg show</code>
<code>rosmmsg list</code>	<code>list all messages</code>
<code>rosmmsg md5</code>	<code>display message md5 sum</code>
<code>rosmmsg package</code>	<code>list messages in a package</code>
<code>rosmmsg packages</code>	<code>list packages that contain messages</code>

TOPICS

One of the most important communication concepts used in ROS and also in the NRP are topics. As already mentioned, topics are used to exchange messages by allowing nodes to publish messages on certain topics and subscribe to these topics to get the associated messages. This decouples the information production from its consumption. Each topic has the type of the ROS messages which are published to it. All subscribers can only receive messages if the type matches, however, type consistency among the publishers is not enforced. [For14; Wad16]

To display debug information of ROS topics the command-line tool `rostopic` is provided. According to the official Wiki [Wad16], the following commands are supported at the moment:

<code>rostopic bw</code>	<code>display bandwidth used by topic</code>
<code>rostopic delay</code>	<code>display delay for topic which has header</code>
<code>rostopic echo</code>	<code>print messages to screen</code>
<code>rostopic find</code>	<code>find topics by type</code>
<code>rostopic hz</code>	<code>display publishing rate of topic</code>
<code>rostopic info</code>	<code>print information about active topic</code>
<code>rostopic list</code>	<code>print information about active topics</code>
<code>rostopic pub</code>	<code>publish data to topic</code>
<code>rostopic type</code>	<code>print topic type</code>

2.1.3 GAZEBO

Gazebo is a 3D multi-robot simulator, which is compatible with ROS. It is used in the NRP to simulate the robots as well as the world with its sensory-rich environment. In addition to that, it also provides information about the scene and the associated models and enables model creation. It is able to simulate indoor and outdoor terrains on high-quality graphics through the open-source graphics rendering engine OGRE3D. Moreover, it provides access to multiple physics engines like Open Dynamics Engine (ODE), Bullet, Simbody and Dynamic Animation and Robotics Toolkit (DART) for

2.1 THE HUMAN BRAIN PROJECT (HBP) AND THE NEUROBOTICS PLATFORM (NRP)

dynamics and kinematics simulation of rigid bodies. To enrich the environment and empower the robot to react to changes, Gazebo provides the possibility to generate sensor data from 2D/3D cameras, laser range finders and a variety of sensors. Another very important feature of Gazebo is that it provides the possibility to run the simulation on a remote server and connect to it over TCP/IP. However, like ROS, Gazebo only runs on Unix-based platforms. [Gazndc; Rosndb]

Gazebo uses different kinds of Simulation Description Format (SDF)-files to represent the individual parts of a scene. World files represent the highest level and include a description of the entire environment. They usually have the extension `.world` and include different models and sensors. Thereby, the term "model" describes all objects with a physical representation like robots, lights, static objects and so on. Every model has to contain at least one rigid body and may contain some joints and sensors if necessary. Thereby, sensors provide data to the model and the simulation, whereas joints are used to connect rigid bodies with each other and perform motion if a force is applied. The subdivision of models in separate files encourages model reuse and allows for a clearer structuring of the world file.

To populate the world, Gazebo needs to parse the world file. This is done with the executable program `gzserver`, which uses the physics and sensor engine to simulate the world. Since Gazebo uses a distributed architecture, it includes all engines, communication mechanisms and visualization parts as libraries. Especially, it contains the communication library, the physics library, the rendering library, the sensor generation library and the GUI library. The physics and rendering libraries use the engines mentioned above, whereas the communication library uses Google Protobuf for serialization of messages and `boost::ASIO` for the communication management. Sending and receiving messages is performed via publishers and subscribers, which use topics as a medium. To enable all libraries to work together and know each other, Gazebo also provides a master which serves as a nameserver.

The visualization of the GUI and the interaction with the simulation is handled by the graphical client which can be executed using the command `gzclient`. [Gaznda; Gazndb; Koe04]

TOPICS

Like ROS, Gazebo also uses topics for communication. The concept works exactly as the one mentioned in section 2.1.2 and allows different publishers to send information to the subscribers by using the topic as a medium.

In contrast to ROS, Gazebo has a fixed set of topics which are defined via `.proto`-files.

These topics usually provide information on the scene, the light, the different models, their poses and much more.

To get information about these topics, the command `gz topic` can be used with the following parameters as provided by the command-line help message (retrieved via `gz topic -h`):

```
gz topic -w [--world-name] arg           World name
gz topic -l [--list]                     List all topics
gz topic -i [--info] arg                 Get information about a topic
gz topic -e [--echo] arg                 Output topic data to screen
gz topic -v [--view] arg                 View topic data using a QT
                                          widget
gz topic -z [--hz] arg                   Get publish frequency
gz topic -b [--bw] arg                   Get topic bandwidth
gz topic -u [--unformatted]              Output state from echo without
                                          formatting
gz topic -d [--duration] arg             Duration (seconds) to run.
                                          Applicate with echo, hz, and bw
```

2.1.4 UNITY3D-CLIENT

The Unity3D-Client is used as a base project for this thesis and should be briefly explained here to distinguish the previously accomplished work from the contributions of this thesis. The client was developed by Sandro Weber and is based on the RoboyVR project ¹. It implements the connection to the NRP as well as the rendering of the virtual environment. It also implements all functions needed for the topic-based communication with the platform, including general subscriber, publisher and advertiser implementations. Moreover, the client supports the HTC Vive to view the simulation.

CONNECTION TO THE HBP

The client connects to the NRP via web sockets. For the communication between the client and the server, the topic concepts of ROS (cf. section 2.1.2) and Gazebo (cf. section 2.1.3) are used.

For a better understanding of the client's functionality, a sequence diagram of the already implemented code for the connection establishment is attached in appendix A.1.

¹<https://github.com/Roboy/RoboyVR>

As the diagram shows, two connections with the server need to be established. One connection on port 8080 to the Gazebo bridge and another one on port 9090 to the ROS bridge. Once the connections are opened, the client is able to receive messages from all subscribed topics and publish messages to all available topics.

As already mentioned in section 2.1.3, the topics used by Gazebo are predefined and include information on the scene and all models used in it. Hence, these topics are used to recreate the scene in Unity and instantiate the correct models. Thus, messages from these topics arrive right after the connection to the NRP is established and are passed to the renderer, who instantiates and updates the models. In contrast to that, the topics used by ROS are specific to the NRP and cover specialized cases like the avatar rotation and velocity or the state of the robot's wheels. As these topics are not predefined, the client does not implement any ROS communication by default.

VISUALIZATION WITH THE HTC VIVE

To allow the user to see the simulation in VR, the HTC Vive and the SteamVR plugin of Unity are used. In the original client, the SteamVR object is placed in the center of the scene and does not move. This allows the user to look around and move in the scope of the HTC Vive's predefined play area. However, it is not possible to go further than this area, which restrains the user from exploring the whole scene. Thus, this aspect needed to be changed in the course of this thesis (c.f. section 4.2.2).

2.2 VIRTUAL REALITY (VR)

The very first idea of VR goes back until 1965, where Ivan Sutherland presented the idea of the "Ultimate Display" with the following words:

"The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal. With appropriate programming, such a display could literally be the Wonderland into which Alice walked." [Sut65]

Ivan Sutherland presented a daring vision of a virtual world where people would not only be able to hear and interact with objects but also to smell and taste them. A few years later, he proved his vision right by creating "The Sword of Damocles", the first Head Mounted Display (HMD) system with head tracking and stereo vision. [Maz99]

Many years have passed since this first step and although we still have a long way to go before we reach the "Ultimate Display", technology has advanced far enough for consumers to be able to buy VR headsets on the market and use them to immerse in a virtual wonderland, just like Ivan Sutherland envisioned so many years ago. [Bis92]

Although, VR is well known these days and a lot of researchers occupy themselves with improving it and getting closer to Sutherland's vision, finding a universally valid definition to describe VR with all its possibilities and limitations is quite difficult. As no single "true" definition can be found, the following definitions were chosen as some representatives to illustrate what points they all have in common and to which characteristics VR can be narrowed down to.

- "Virtual reality a medium composed of interactive computer simulations that sense the participant's position and actions and replace or augment the feedback to one or more senses, giving the feeling of being mentally immersed or present in the simulation (a virtual world)." [She03]
- "Real-time interactive graphics with three-dimensional models, when combined with a display technology that gives the user immersion in the model world and direct manipulation [...]" [Bis92]
- "Virtual reality is a high-end user-computer interface that involves real-time simulation and interactions through multiple sensorial channels. These sensorial modalities are visual, auditory, tactile, smell, and taste." [Bur03]
- "Virtual reality lets you navigate and view a world of three dimensions in real time, with six degrees of freedom (6DOF). [...] In essence, virtual reality is a clone of physical reality. In physical reality, you [...] experience real time, and have the ability to interact with the world around you. VR products mimic those conditions, [...]" [Von95]

Although, these definitions show some differences, they agree on the fact that VR is an interface, which enable the user to interact with a virtual world and immerse in it. These characteristics are also reflected in the four key elements of VR proposed by Sherman and Craig [She03] or in the three I's of VR proposed by Burdea and Coiffet [Bur03].

The first element, Sherman and Craig introduce, is the virtual world which is "an imaginary space often manifested through a medium" [She03]. As the definition already states, this world is imaginary and can only be shared with others through a medium like VR. Thus, imagination is necessary to create the world and also to engage in it and all it's possibilities. This brings us to the first "I" of VR, the imagination. Another very important characteristic of VR is the interaction with the virtual world, which is

represented as a key element by Sherman and Craig as well as an "I" by Burdea and Coiffet. It describes the response of the system to different actions performed by the user. In case the virtual world and the interaction with it is done well the user is able to immerse in the experience and get a sense of presence in the virtual world. As immersion differentiates a good from a bad VR experience, it represents the third key element and final "I" in the two taxonomies. The last key element proposed by Sherman and Craig is the sensory feedback, which describes the feedback provided to the user based on his physical position or action. This element is not part of the taxonomy proposed by Burdea and Coiffet as it can also be counted to interaction. [Bur03; She03]

2.2.1 THE HTC VIVE

The HTC Vive is a VR system developed by Valve and takes VR closer to the original idea of Ivan Sutherland. It allows the user to walk around in a predefined play area of maximal 5 m in diagonal and thereby control the movement of his virtual self just by moving in reality. The tracking is done via sensors in the headset and the two base stations, which can track the headset in the play area, independently of its orientation. The distance of the base stations defines the play area, which forms a virtual boundary for the HTC Vive. The headset itself has an OLED display with a resolution of 1080 x 1200 pixels per eye, a refresh rate of 90 Hz and 110° field of view. It uses SteamVR Tracking, a G-sensor, a gyroscope and proximity sensors to position the head of the player correctly in virtual space. [HTCnd]

As the headset uses SteamVR Tracking, the whole system can be easily integrated into Unity just by downloading the SteamVR plugin ¹ from the Unity Asset Store.

The HTC Vive was chosen for this thesis out of two reasons. First, because the communication with the HTC Vive was already implemented in the Unity 3D-Client. And second, because the supported trackers allow for a great variety of interaction techniques.

2.3 GESTURE-BASED INPUT

As VR has become more popular over the last couple of years, the need to overcome the inconvenience of traditional Human-Computer Interaction (HCI) tools like mouse and keyboard has switched into the focus of many researchers. One alternative to these

¹<https://assetstore.unity.com/packages/templates/systems/steamvr-plugin-32647>

methods are gesture-based inputs, as users are accustomed to utilize gestures to express their feelings and thoughts. Thus, using gestures to transfer commands to the computer feels quite intuitive and has been used in several multimodal interfaces to improve the way of communication. [Kim00; Cab05]

A good definition of gestures has been given by Hummels et al. who states that "a gesture is a movement of one's body that conveys meaning to oneself or to a partner in communicatio" [Hum98]. In this context, gestures may include all kinds of movements including body, face, eye and hand movements. However, most VR applications focus on body and hand movements to track the user and his movements in virtual space. [Hum98]

Nielsen et al. pointed out that "a gesture interface should by nature be ergonomically superior to physical handheld devices, which introduces external force" [Nie04] if main ergonomic principles like the inter-finger constraints are taken into account. However, performing gestures over a longer period of time can become physically stressful nonetheless. [Nie04; Cab05]

2.3.1 THE MYO

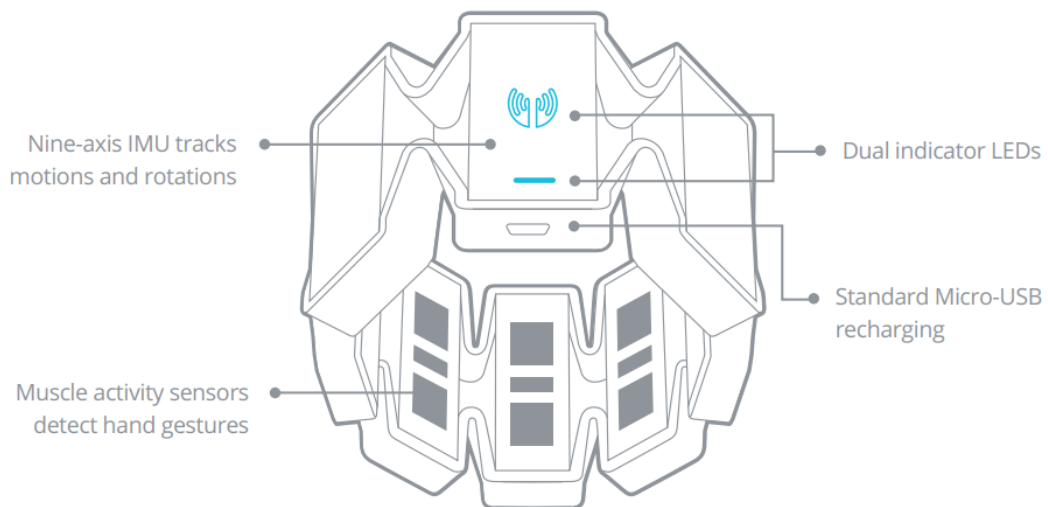


FIGURE 2.4: Overview of the Myo with its sensors and expandable casing from [Thandd].

The Myo is a gesture-recognition armband developed by Talmic Labs. It uses eight Electromyography (EMG) sensors depicted in figure 2.4 to recognize different hand gestures. Therefore, it needs to be put on the thickest part of the user's forearm to be able to read the neuromuscular activities. Every time a muscle is contracted, an electrical current is generated in this muscle which is then measured by the EMG sensors.

Reaz et al. states that "the EMG signal is a complicated signal, which is controlled by the nervous system and is dependent on the anatomical and physiological properties of muscles" [Rea06]. As every person has slightly different muscles, the signal also differs from person to person which makes the gesture classification quite difficult. [Thandb; Rea06]

In addition to the EMG sensors, the Myo also contains a highly sensitive nine-axis Inertial Measurement Unit (IMU), which provides movement and orientation data in form of pitch, yaw, and roll of the user's arm. The data is acquired through the three-axis gyroscope, the three-axis accelerometer, and the three-axis magnetometer. This spatial information combined with the predefined gestures depicted in figure 2.5 create new interaction possibilities. To increase the number of possibilities, the gesture data is also provided as raw EMG-data which allows developers to define their own gestures or improve the gesture recognition rate by applying different machine learning approaches. [Thandb; Thandd; Thandg]



FIGURE 2.5: Myo gestures from left to right: Wave left, wave right, double tap, fist, fingers spread. Taken from [Thandf].

Another feature of the Myo is, that it is able to recognize on which arm it is worn, which way it is oriented and when it is removed from the user's arm. To gather this information the Myo requires the user to perform a synchronization gesture, in the form of the wave right depicted in figure 2.5, whenever the user puts on the armband. If the gesture is correctly recognized, the Myo vibrates, to indicate a successful synchronization. This vibration represents another important feature of the Myo, namely the possibility to give haptic feedback through three different vibration types: short, medium and long vibrations. [Thandb; Thandg]

The hardware specifications of the armband show that the Myo uses an ARM Cortex M4 Processor and Bluetooth 4.0 LE for the communication with the target platform. It supports Windows 7, 8 and 10, Mac OS 10.8 (Mountain Lion) and above, iPads with iOS 7.0 and higher and Android devices with Android 4.3 (Jelly Bean) and higher. It also provides a Unity SDK for Windows and Mac OS which can be used in combination with Myo Connect a software from Thalmic Labs, which guides the user through the first steps, deals with the connection establishment between the Myo and the target platform, provides the possibility to create your own profile and enables Myo's presentation control and other features. [Thandg; Thanda]

CHAPTER 3

RELATED WORK

This chapter covers some related work and research. This research covers the area of gesture-based avatar navigation in VR and work focusing on the Myo armband. The part focusing on the Myo armband, comprises the usage of the Myo in VR as well as for navigation. Thereby, only one research combines both application areas.

3.1 GESTURE-BASED AVATAR NAVIGATION IN VR

Since VR has become popular, we face the challenge of interaction and movement in virtual environments as traditional input methods are not suitable for this. Thus, many researchers have dedicated themselves to this topic.

Lee et al. [Lee98] developed a continuous hand gesture recognition system which was combined with a motion database to allow the user to control the basic motions, which are depicted together with the appropriate hand postures in figure 3.1. The different postures were inspired by the Korean Sign Language and can be combined with hand movements to further specify the motion. For example, a movement to the front can be triggered by making the sign for walking and pushing the hand forward at the same time. Pulling the hand with the same sign backward would lead to a backward movement and the same principle holds for right and left movements. [Lee98]

For the data acquisition, Lee et al. used the CyberGlove and the Polhemus Fastrak system. This data is then forwarded to direction and posture classifiers which recognize the different gesture attributes and allow the following interpreter stage to classify the gesture. The gesture is then translated into the corresponding movement, whereas a

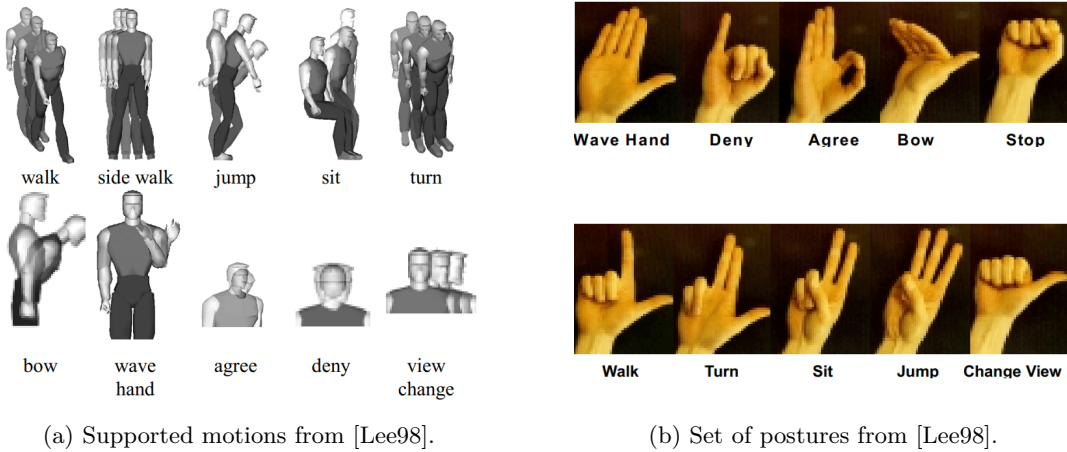


FIGURE 3.1: The left picture show the ten motions supported by the system and the right picture shows the corresponding hand postures needed to trigger these motions.

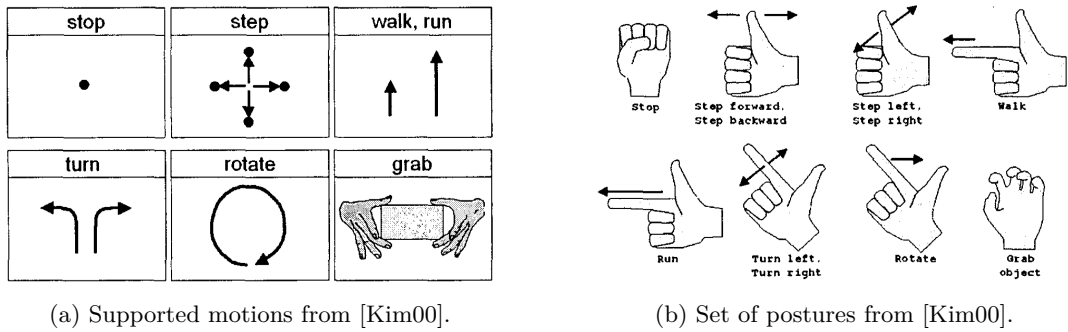


FIGURE 3.2: The left picture shows the eleven motions supported by the system and the right picture shows the corresponding hand gestures needed to trigger these motions.

state transition function is used to determine which movement is possible considering the current state. [Lee98]

As the research of Lee et al. primarily focused on the correct classification of hand gestures, it provides no data on the user experience. However, given the complexity of the gesture set, it can be assumed, that it is quite difficult for untrained users to utilize this method. In addition to that, the proposed gestures put a lot of strain on the user’s hand, which eventually would lead to fatigue and pain in the hands.

Kim et al. [Kim00] used a very similar approach a few years later when they developed a movement method based on only four, instead of ten hand postures. These postures in combination with the four different hand motion-directions forward, backward, left and right, allow the user to perform eleven different motions as depicted in figure 3.2a. A short comparison of figure 3.1 and figure 3.2 shows, that the method proposed by

3.1 GESTURE-BASED AVATAR NAVIGATION IN VR

Kim et al. pass on special, rarely used motions like jump, sit, bow, wave hand, agree, deny and view change. Instead, it provides a small, easy to remember set of gestures, which not only cover basic movements but also running and additionally the grabbing of objects. [Kim00]

They used an improved version of the CyberGlove, the CyberTouch system to acquire the joint-angle measurements of the fingers and the Fastrak system to acquire the hand movements. As well as Lee et al. [Lee98], they also use a direction and posture recognition module to identify the performed gesture. The final avatar movements are then acquired using a finite state automata. [Kim00]

This research again focuses primarily on the gesture detection part, and provide no information on the usability of the method. However, it can be assumed, that the smaller gesture set increases the learnability compared to the technique proposed by Lee et al.

Going away from CyberGloves and expensive tracking systems, Şen et al. [Şe12] uses the motion-tracking capabilities of the Microsoft Kinect to develop a gesture-based navigation method for the already existing VR simulation called "Re-discovering Vrouw Maria", which shows the wreck of a ship underwater. The proposed navigation method uses a pointing-based gesture set where the user goes in the direction he is pointing at as long as he continues pointing. To acquire the pointing-direction the skeleton tracking system of the Kinect is used. Thereby, the pointing direction is the vector which goes from the shoulder to the hand. In contrast to the previously mentioned works, this method also supports velocity control. For this purpose the velocity is mapped to the distance between hand and body, the farther the hand is away, the faster the movement is. In addition to that, the proposed method also supports object selection which is triggered by a collision between the viewpoint and the activation zone of the desired object. [Şe12]

A user study performed by Şen et al. state, that the advantage of the pointing-based method is the simple vocabulary, which has a low error rate, is easy to learn and remember and comes with relatively low physical stress. [Şe12]

Another method proposed by Codd-Downey et al. [Cod14] uses the Leap Motion to implement a pointing-based technique for the navigation control in 3DOF. As figure 3.3 shows, four hand gestures are needed to move forward, reverse, turn and stop. To make the turning easier, not a direct, but a relative mapping from the absolute measurement of the finger to the rotation angle was used. The speed can be controlled over the yaw of the user's index finger.

Compared to the other techniques, this one is less mighty, as it doesn't support special movements like stepping to the side. However, Codd-Downey et al. state, that the incor-



FIGURE 3.3: Overview of the gestures used in the method proposed by Codd-Downey et al. [Cod14]. From left to right: forward, reverse, turn and stop.

poration of additional capabilities like jumping could be easily added without changing the gesture set, just by using the horizontal and vertical translation of the hand. [Cod14]

One of the most recent works in the field of gesture-based navigation was published by Zhang et al. [Zha17] who declared, that the greatest weakness of single hand navigation methods is that they "do not provide a way for a user to turn while walking" [Zha17]. Thus, they propose a technique which uses both hands for navigating. Like Codd-Downey et al. [Cod14], they also used the Leap Motion to track the hands' movements. However, their gesture set depicted in figure 3.4 is much larger and allows for a greater variety of movements, including the possibility to turn left while walking forward which is depicted in the last image of figure 3.4b. As the images show, moving and turning are controlled by the left and right hand respectively. While turning, the speed stays constant at $15^\circ/\text{s}$. According to Zhang et al. this should allow the movement to become more seamless and help the user to feel more comfortable. [Zha17]



(a) Hand gestures for movement, from left to right: walking forward, running forward, walking backward, running backward, stop. Images from [Zha17].

(b) Hand gestures to turn, from left to right: turn left, turn right, stop turning and a combination of both hands showing how to turn left while walking forward. Images from [Zha17].

FIGURE 3.4: Gesture mapping of movements and turns to the left and right hand.

In contrast to the other papers, this one provides a user study, which compares the proposed method with an Xbox controller and the Oculus Touch controller. Thereby, the users' task was to get through two maze scenes, whereas one of the scenes was used as a training area to let the users get used to the technique. The different methods were compared with regard to different time measurements. From the acquired results, Zhang et al. state, that most participants preferred the newly proposed technique over the others, although they were faster with the controller. They conclude, that their method causes low fatigue and is easy to use and learn.

3.2 RECENT WORKS ON THE MYO

Since the Myo is a relatively new device, there only exist a few scientific works covering its usage for VR or navigation tasks.

3.2.1 MYO IN VR

A very recent master thesis from Bonome [Bon17] focuses on the Myo and its capabilities as an interaction technique for virtual environments. The main research question of the work is: "Has MYO armband high grade of usability for manipulation of 3D objects in Virtual Reality Environments?" [Bon17]. To answer this question, the first part of the thesis consists of a literature review covering the Myo with its advantages and limitations. Besides this, the thesis develops two interaction methods, the Soft-Grab and the Hard-Grab technique. Both approaches are used to select and manipulate objects in virtual environments, whereas the selection of the objects is always performed by pointing at the object with the Myo and making a fist. The only difference between the two methods is, that for the Hard-Grab technique the strength used to make a fist must be proportional to the weight of the object one wants to select. After the object is selected, the user can move it around through movements of his arm. However, it is important to note, that the user must maintain the fist gesture during these movements, as the object is deselected when the fist is released. [Bon17]

A comparison showed that both methods have high usability grades, which demonstrate, that the Myo can be used as an interaction technique. In addition to that, Bonome claims that the possibility to measure the strength applied to a gesture and the vibration feedback system allow for richer user experiences. However, he also states, that users reported problems with muscle fatigue during the experiments, which has to be analyzed in further studies.

3.2.2 MYO FOR NAVIGATION

In his master thesis, Patel [Pat15] used the Myo to navigate the semi autonomous, smart wheelchair Anna, which is equipped with a variety of different user interfaces to provide appropriate control mechanism for each type of user. The wheelchair consists of four wheels, which allow the user to drive forward and turn around. For the wheelchair navigation, four standard gestures of the Myo are used. The fist is used to move forward, whereas the finger spread gesture is used to stop the movement. Turning right and left is controlled via wave right and wave left respectively. [Pat15]

The corresponding tests showed, that the error rate of the fist gesture is 9 % and the

one of the finger spread gesture is 8 %. Thus, they have a higher error rate than the wave right (6 %) and the wave left (4 %) gesture. However, it is not stated if these measurements were taken with a calibrated Myo, so it is still unclear if these results can be improved by calibrating the Myo. [Pat15]

Another work covering the utilization of the Myo to navigate a mobile robot is from Luh et al. [Luh15]. They propose two different navigation techniques for an omnidirectional robot. The first technique works identically to the one of Patel [Pat15] with the difference that the fist is used to stop the robot and the finger spread gesture is used to move the robot forward. However, the problem with this approach is that it doesn't utilize the whole capabilities of the omnidirectional robot as it is not possible to rotate and translate the robot at the same time. To solve this problem, the second approach not only uses the gesture data of the Myo but also its spatial data. They make it possible to move forward and rotate the robot at the same time through holding the fist and turn it $\pm 90^\circ$ around the roll-axis. According to their experiments, this approach also provides the advantage of smoother movements. [Luh15]

Their pilot study on the gesture recognition rate of the Myo show, that the gestures of the Myo are recognized very well with 98.17 % of recognition for the fist, 95.68 % for the spread fingers, 99.45 % for the wave right and 99.82 % for the wave left gesture. Unfortunately, no recognition rate for the double tap gesture is provided. However, it is stated, that this gesture is easily misclassified. [Luh15]

3.2.3 MYO FOR NAVIGATION IN VR

McCullough et al. [McC15] were the first to explore Myo's capabilities as a gesture-based navigation approach for VR as it is inspired by this thesis. They designed a method which allows the user to move into the viewing direction through swinging the arms. Thereby, the user's orientation is controlled through physical rotations, which are tracked through the sensor on the Oculus Rift DK2. Their goal was to find an inexpensive method which allows the user to freely explore the virtual environment and thereby support the task of spatial orientation. To evaluate the suitability of the arm-swinging method, it was compared against a wireless joystick (Logitech Freedom 2.4) and a walking-based method. The walking-based approach consists of two LEDs mounted on the HMD which are tracked by eight optical cameras. The position data is then interpolated by the WorldViz PPT software. [McC15]

The three methods were evaluated in terms of their mean turning error, their mean latency, and their user experience, whereas the results of the user experience evaluation are shown in table 3.1. The evaluation results show, that the proposed method performs

3.2 RECENT WORKS ON THE MYO

Question	Joystick	Myo	Walking
Did your movements feel accurately translated into the environment?	4.16	3.33	4.08
Did you experience any motion sickness or vertigo while using this equipment?	1.91	1.42	1.67
Overall, how would you rate your experience with this machine?	3.75	4	4.17
If this device was within your price range, would you buy it?	3.08	3.17	3.5
If you owned this device, how frequently would you use it?	2.92	2.75	3.25

TABLE 3.1: Table from McCullough et al. showing the mean responses of the users on a 5-point Likert scale, where 5 indicates high agreeableness. [McC15]

better than the joystick and equally well as the walking-based method. When using the walking-based method or the arm-swinging method users had an improved spatial awareness in the virtual environment compared to the joystick. McCullough et al. conclude, that the Myo is well suited for exploring virtual environments. However, they also mention some drawbacks of their method including the missing possibility to go backward, which was taken into account when designing the navigation methods for this thesis. [McC15]

CHAPTER 4

COMPARISON OF DIFFERENT GESTURE-BASED NAVIGATION APPROACHES

Since gesture-based navigation methods are not as established as controller-based ones, a lot of different approaches need to be tested to find a satisfying one. Therefore, we developed different navigation approaches, which were compared against each other in a preliminary user study.

The whole process from the development of the different methods, over the implementation to the final tests is covered in this chapter, which is structured as follows: Section 4.1, covers the development of different gesture-based navigation methods. Therefore, we first analyzed what motions are substantial for a good navigation method and how those can be realized with the available technology. Afterward, the different methods were implemented in Unity as shown in section 4.2. This section also includes the utilized technologies and their specifications. Finally, section 4.3 includes the preliminary user study which compared the different navigation methods against each other as well as the goals of the study and its setup. The chapter is completed with the discussion of the study results in section 4.4.

4.1 APPROACH

To develop a good and easy to use gesture-based navigation method, a review of the possibilities provided by the different interaction technologies is needed. A clear definition of the desired movements is also necessary. The technological possibilities can be summarized as follows: The HTC Vive is capable of tracking the head as well as the

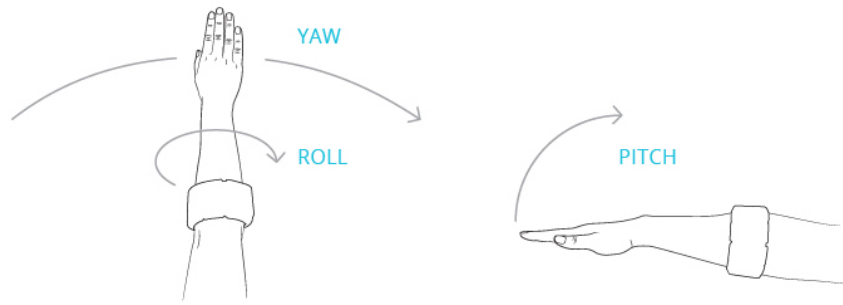


FIGURE 4.1: Break down of the Myo's orientation into yaw, roll, and pitch for an easier understanding. [Mar14a]

body, whereas, the Myo provides spatial data and the recognition of five gestures. The spatial data includes the acceleration of the Myo as well as its orientation, which can be broken down into yaw, roll, and pitch as depicted in figure 4.1. [Thandb]

The desired movement control mechanism include the starting and stopping of a motion, the control of the movement's speed and direction as well as the control of the user's rotation in virtual environment.

4.1.1 CONTROLLING THE BEGIN AND END OF A MOTION

We use the Myo in three different ways to control the starting and stopping of movements.

The first way is called "StartStop" and uses a gesture to initiate a forward motion. This motion continues until the gesture is made again or another dedicated stopping gesture is performed. Since the Myo only supports five gestures it was decided to use the same gesture for starting and stopping as the other gestures might be needed in the future to perform additional motions not implemented yet.

The advantage of this method is, that after the motion's initialization, the user doesn't have to do anything to continue the movement. This is especially useful for traveling longer distances as the user can just relax during these forward movements. However, this advantage is also a drawback, which becomes palpable when moving very short distances. Since the motion continues until the gesture is repeated, very short movements require the user to successively perform the same gesture twice, in a very quick manner. As most backward movements are done to navigate out of objects, these motions are typically quite short and therefore, face this problem. Because of this, no backward movements were designed for this method.

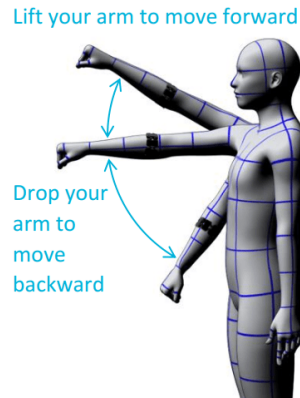


FIGURE 4.2: Illustration of the "ArmPitch" method, showing that raising the arm leads to a forward movement and lowering the arm to a backward movement.

The second way of motion control is referred to as "HoldToMove" method, which again uses a gesture to start moving. To keep moving the user needs to hold the gesture. This motion stops as soon as the gesture is released. In contrast to the first method, this one allows an active motion-control as the user needs to actively do something to continue the movement. In addition to that, this method allows an easy inclusion of backward motions just by adding an additional gesture.

The third way of motion control, which is referred to as "ArmPitch" method, uses the pitch of the user's arm to initiate a movement. Thereby, the users' arm is lifted from its initial position to move forward and lowered, to move backward as shown in figure 4.2. The movement continues as long as the arm is not moved back to its initial position. To allow forward and backward movements the initial position of the arm was chosen to be horizontally outstretched in front of the user.

To decide which gesture should be used for the "StartStop" and "HoldToMove" methods, the research from Patel [Pat15] and Luh et al. [Luh15] was analyzed. As they do not recommend the double tap gesture we are left with the fist, the finger spread, the wave left and the wave right gestures. Although the last two gestures have better recognition rates, they feel very unnatural and especially the wave right gesture puts a lot of stress on the user's hand. Since the fist gesture exhibits slightly better recognition rates than the finger spread gesture, it is used for both methods. For the "HoldToMove" method we also use the finger spread gesture for backward movement as this motion is not as frequently used as the forward movement.

4.1.2 CONTROLLING THE MOVEMENT'S SPEED

When controlling the speed, it can be chosen between continuous and step-wise speed control. As the proposed navigation method should be as intuitive as possible, a continuous speed control is desired. To implement this, we can either use the pitch or the roll of the user's arm. As the arm's pitch has a much higher range and feels more comfortable while using it, this method is chosen as speed control mechanism.

4.1.3 CONTROLLING THE MOVEMENT'S DIRECTION AND USER'S ROTATION

The movement direction can be either controlled through the HTC Vive or the Myo. When using the HTC Vive, called "ViveDirection" method, the direction of movement is defined through the head-tracking system and therefore corresponds to the viewing direction of the user. Hence, viewing and moving direction are coupled to each other which disallows the user to look around while moving.

The direction control with the Myo uses the spatial data which is provided by the armband. It utilizes the yaw of the user's arm, whereby the moving direction is the user's pointing direction.

When defining the user's rotation as well as the viewing direction through the HTC Vive, and the moving direction through the Myo the user is able to freely look around while moving in a specific direction. This method is called "MyoDirection" method. When using the Myo instead of the HTC Vive to control the user's rotation and viewing direction, called "MyoRotation", the user is not able to freely look around anymore while moving.

4.1.4 RESULTING NAVIGATION METHODS

By combining the different control mechanism from section 4.1.1 with the direction and rotation control mechanism from section 4.1.3, the following navigation methods assemble. All methods utilize the speed control mechanism described in section 4.1.2.

- "StartStop & ViveDirection"
- "StartStop & MyoDirection"
- "StartStop & MyoRotation"
- "HoldToMove & ViveDirection"
- "HoldToMove & MyoDirection"

- "HoldToMove & MyoRotation"
- "ArmPitch & ViveDirection"
- "ArmPitch & MyoDirection"
- "ArmPitch & MyoRotation"

4.1.5 VIBRATION FEEDBACK

When working in the field of VR, the challenge of providing information to the user without showing them on the screen quickly arises. One especially good and simple method to meet this challenge is by using vibration feedback.

For our methods, two different types of vibration feedback are used. The first type is a single vibration which is used to notify the user that a gesture was recognized by the Myo. The second type of feedback is called pulsing, which is a short, regularly repeating vibration. This feedback mechanism is used to signal the player that he is currently moving.

This is especially important for the "StartStop" methods, as the user might otherwise not realize that he is still moving. It is also important for the "ArmPitch" methods as the user needs some kind of help to regain the initial position.

According to Riecke et al. [Rie05], vibrations also facilitate the illusion of self-motion and make the motion feel more realistic. The research of Plouzeau et al. [Plo15] also suggests that proprioceptive vibrations, which are activated when the user moves, have a good impact on the motion sickness. Although, the vibrations they used are applied to the main extensor muscle of the hip and the vibrations used in this thesis are applied to the muscles of the upper part of the forearm, it may still help in reducing the motion sickness, as normal human movements also cause slight vibrations in the human body. [Plo15]

4.2 IMPLEMENTATION

For the implementation of the different navigation methods, the tools and technologies described in section 4.2.1 were used. The implementation itself was done with C# and is described in section 4.2.3.

4.2.1 UTILIZED TOOLS AND TECHNOLOGIES

This section presents a short overview of the utilized tools and technologies. Thereby, table 4.1 shows an overview including the model numbers, versions and specifications.

Computer Specifications	
Processor	Intel(R) Core(TM) i7-6700K
RAM	16 GB
OS	64-bit Windows 10 Education
Graphics Card	NVIDIA GeForce GTX 970
Unity	
Version	2017.1.1f1
Microsoft Visual Studio Community 2017	
Version	15.4.5
HTC Vive Headset	
Screen	Dual AMOLED 3.6" diagonal
Resolution	1080x1200 pixels per eye
Refresh rate	90 Hz
Field of view	110 degrees
SteamVR	
Version	1507941678
SteamVR Plugin for Unity	
Version	1.2.1
Myo	
Firmware Version	1.5.1970
Model	MYOD5
Part Nr.	MYO-00002-001
Myo Connect	
Version	1.0.1
Myo SDK	
Version	0.9.0 for Windows
LimeSurvey	
Version	2.72.5+171121

TABLE 4.1: Utilized tools and technologies with the appropriate versions and specifications.

Unity: Unity is a development platform for games and was used to create an environment to test the different navigation methods. It was chosen out of two reasons. First, it is a supported platform of the HTC Vive, connected with the SteamVR plugin, and the Myo, connected with the Myo SDK. Hence, the incorporation of both technologies into the platform is quite simple. Secondly, the Unity3D-Client, which serves as a base-project for the second part of this thesis, is an Unity project itself. [Thandc]

Microsoft Visual Studio: Visual Studio is a development environment developed by Microsoft, which works well with Unity. With IntelliSense and easy code navigation functionalities like "Peek to definition" and "Navigate to" it allows easy and quick C# coding. [Micnd]

SteamVR: According to Valve's programmer Joe Ludwig [Lud16], SteamVR includes everything, from the technology shipped with the HTC Vive, over all APIs to Steam itself running in VR. The suite of API's which is used to interact with the VR system is called OpenVR and this part is also included in the SteamVR plugin for Unity, which was used for the development of the test setup. These APIs provide, among other things, device drivers, an interface to the compositor to display textures on the HMD and access to the Chaperone system configuration. [Lud16]

Myo Connect: Myo Connect is a software comprising several tools for working with the Myo. These include the Armband Manager, which deals with the connection to the Myo armband(s) and administers the calibration profile. [Thande]

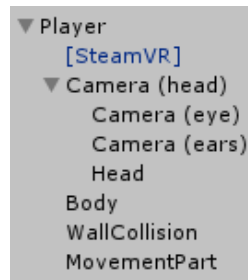
LimeSurvey: LimeSurvey is an open source software used to design and run online surveys. It makes the design of surveys very easy and provides many different question types, analysis possibilities, export options and dynamic questions, which are influenced by previous answers. It was run on a local computer to design and host the questionnaire. [Limnd]

4.2.2 IMPLEMENTATION OF THE AVATAR

To be able to test the different navigation methods, an environment and an avatar is needed. This environment was created using Unity and Visual Studio. As the navigation methods should be used in the NRP, the avatar was designed in accordance with the one in the NRP. Thus, the avatar is kept very simple, with just a sphere as a head and a capsule as body.

The avatar is an object in the Unity scene, called **Player**, which structure is depicted in figure 4.3. To be able to decouple the head- and body-control, the avatar contains child-objects, which comprise the different body representations and scripts.

The avatar includes the SteamVR prefabs required by the HTC Vive to create the VR view. Those are the prefabs [SteamVR] and [CameraRig], whereas the latter prefab is only represented by the child-object **Camera (head)** as the other objects are not needed

FIGURE 4.3: Unity Hierarchy-view of the GameObject `Player`.

for the proposed navigation methods. Thereby, the GameObject `Camera (head)` is responsible for tracking the HMD and hence, will always follow the movements of the user’s head. The GameObject `Head`, which just includes the sphere representing the avatar’s head, is a child-object of `Camera (head)`. Thus, the movement of the avatar’s head is always in accordance with the user’s head-movement and is thereby, decoupled from the different gesture-based rotation and movement control methods.

The representation of the body is a GameObject called `Body`, which is located at the same hierarchy level as `Camera (head)`, and `[SteamVR]`. However, this GameObject not only includes the representation of the body, but also the script `AlignMovementToHead`. This script ensures, that the avatar’s body is always connected to the head, even when the head follows the movements of the HTC Vive. As the head also rotates with the user’s head around its center point, it could happen that the user is looking directly into the capsule representing the body. Since this would cause perplexity and an unrealistic avatar-behavior, the script ensures, that the body slightly moves backward whenever the user is looking down.

As the avatar should behave as expected by the user, it should not be able to move through walls. To achieve this behavior the GameObject `WallCollision` was added to the `Player`. This object comprises a `Capsule Collider`, which encompasses the avatar’s body as well as its head, and the two scripts `AlignMovementToHead` and `StopWhenColliding`. The script `AlignMovementToHead` ensures, that the collider always sticks to the user’s head and the script `StopWhenColliding` prohibits the user to move through walls and ensures that the user bumps-back whenever the avatar collides with one.

The GameObject `MovementPart`, which is also part of the avatar, contains the script `MovementThroughViveAndMyo` which implements the different movement methods discussed in section 4.1. A detailed explanation of this script is presented in section 4.2.3.

4.2.3 IMPLEMENTATION OF THE NAVIGATION METHODS

The different navigation methods listed in section 4.1.4 are implemented as a combination of the `ControlType` and the `NavigationType`. The `ControlType` indicates, which method is used to control the starting and stopping of a motion and therefore has three possible states: "StartStop", "HoldToMove" and "ArmPitch". The `NavigationType`, on the other hand, indicates which method is used to control the user's rotation and movement direction. Therefore, the three methods "ViveDirection", "MyoDirection" and "MyoRotation" are provided.

Avatar-Rotation and Synchronization of Myo and HTC Vive: Depending on the chosen `ControlType`, the script starts with a synchronization and rotation phase. As the rotation of the `GameObject MovementPart` determines the movement direction of the avatar, this rotation always needs to be updated.

If the chosen `NavigationType` is the "ViveDirection" method, the Myo's forward direction is synchronized to the one of the HTC Vive as the latter one is used to determine the movement direction. Therefore, a rotation around the y-axis is calculated and applied it to the `GameObject MovementPart`.

As the methods "MyoRotation" and "MyoDirection" use the Myo instead of the HTC Vive to determine the movement direction, the forward direction of the HTC Vive needs to be synchronized to the one of the Myo. However, the corresponding `GameObject Camera (head)` is controlled through the HTC Vive and thus, cannot be directly rotated or moved. To synchronize the two rotations anyway, the parent object `Player` is rotated instead, which leads to an indirect rotation of the `GameObject Camera (head)`. Since the `GameObject Camera (head)` only determines the viewing direction and not the moving direction, the `GameObject MovementPart` needs to be rotated as well. As the moving direction is determined by the Myo, the `GameObject` is rotated around the y-axis to look into the same direction, the user is pointing at. The number of times, this rotation is calculated is the only difference between the "MyoDirection" and "MyoRotation" control method. As the "MyoDirection" control method decouples the rotation of the HMD from the movement direction, the rotation is only calculated once. For the "MyoRotation" method, on the other hand, the rotation must be calculated whenever the player starts moving, as the viewing and moving direction are coupled to each other.

Gesture-Based Control: Whichever `ControlType` is chosen determines which hand gestures trigger a movement. The triggered motion is either a forward, backward or sideward movement. Thereby, the movement direction is specified using an integer, which is set to 1 for moving forward or sideward and to -1 for moving backward.

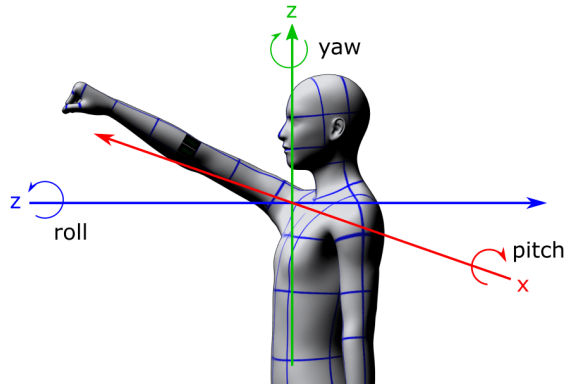


FIGURE 4.4: The Myo's coordinate system in Unity.

The `ControlType` "HoldToMove" triggers a forward movement if the current gesture is a fist. A backward movement, on the other hand, is triggered by the finger spread gesture. The movement stops if the last gesture, detected by the Myo, is different from the current one and the current gesture is neither the finger spread nor the fist gesture. The `ControlType` "StartStop" toggles between starting and stopping a movement whenever the last gesture differs from the current one and the current gesture is a fist. In contrast to the previously mentioned methods the "ArmPitch" method uses the y-component of the Myo's forward vector to determine the type of movement. As figure 4.4 shows, raising and lowering the arm corresponds to a rotation around the x-axis. Thus, the y-component of the movement vector increases if the arm is lifted, and decreases if the arm is lowered. Hence, a y-value over 0.3 is interpreted as a forward movement and a y-value below -0.3 as a backward movement. Every other y-value is interpreted as stopping.

Avatar-Movement: A movement is only triggered if the corresponding gesture is performed. Then, the `GameObject Player` is moved according to the following formula: $position += direction * MovementPart.rotation * speed(y)$. As the formula suggests, the avatar is moved according to the orientation of the `GameObject MovementPart`. Thereby, the variable *direction* is the previously mentioned integer, which is either set to 1 or -1. The applied speed is calculated via a linear function of the form $speed(y) = y * k + d$, whereas, *y* corresponds to the y-component of the Myo's direction vector. Thereby, the function maps the minimal and maximal speed ratio to the arm's minimal and maximal deflection value. For acceleration purposes, the variables *k* and *d* are precomputed at the beginning of the simulation. To attain the vibration feedback, addressed in section 4.1.5, every movement starts an appropriate coroutine, if not already running. This coroutine triggers a short vibration in the Myo which is

repeated after two seconds to create a pulsing vibration. This repetition only stops when the user stops moving.

4.3 EVALUATION

This section describes the goals and structure of the first user study, with the title: "User Study for Different Navigation Approaches in VR".

4.3.1 GOALS

As only one gesture-based navigation method should be implemented into the Unity3D-Client, the goal of this evaluation was to find out, which navigation method performs best. Another aim of this evaluation was to gather as much feedback and ideas for improvements as possible. This should help us to refine the selected method to get closer to the user's expectations. Moreover, the evaluation should help to discover the weaknesses and strength of the Myo as a navigation technology.

4.3.2 PRESELECTION

Before starting with the real evaluation, the nine methods proposed in section 4.1.4 were subjected to a self-test. During this self-test, it became evident, that the control mechanism "MyoRotation", caused severe motion sickness. This might be because a person is not able to hold his arm as steady as his head, which leads to constant movements in VR, which in turn induce motion sickness. As a consequence, it was decided to remove the three methods which use this technique, from the user study. Thus a falsification of the other methods' results, which could occur due to an increased motion sickness caused by the "MyoRotation" method, can be prohibited.

Hence, only six methods were presented to the users.

4.3.3 SETUP AND STRUCTURE

To evaluate the different gesture-based navigation methods, the technologies and tools described in section 4.2.1 were used. The study took place in the Augmented-/Virtual-Reality Lab of the chair of Human-Machine Communication at the Technical University of Munich (TUM). Thereby, the exercises were performed on a dedicated HTC Vive computer and the questionnaire was shown on a separate laptop.

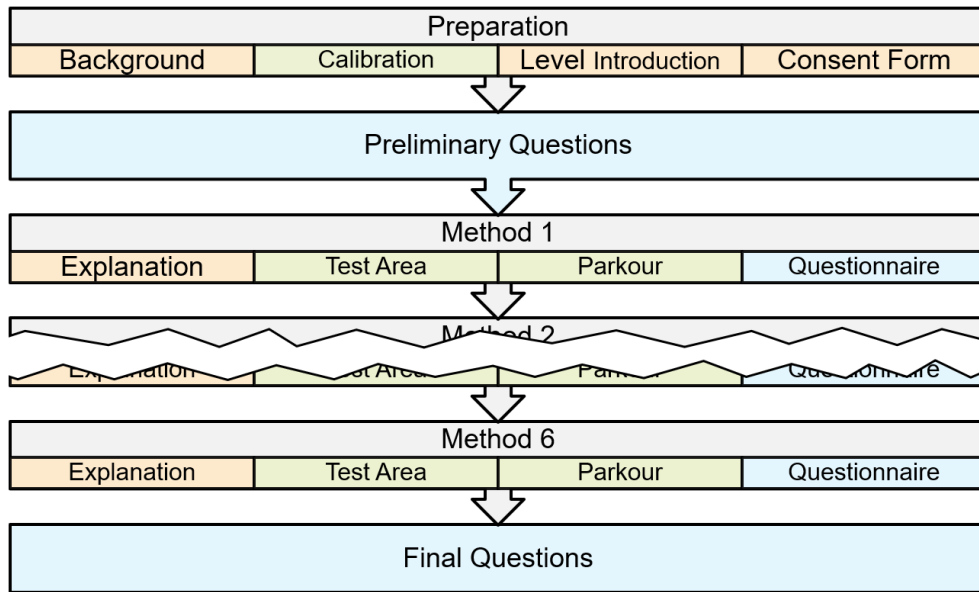


FIGURE 4.5: A schematic overview of the evaluation procedure.

The user study consisted of a practical part, a questionnaire and the observations of the investigator. Thereby, the study itself followed a repeated measures design. As this design requires each user to test all methods, fewer users are needed, which was the reason why this method was chosen. To prevent a bias caused by the order in which the users test the different methods, the order was randomized. The independent variables of this evaluation are the different control methods which are composed of the `NavigationType` and the `ControlType`. The dependent variables captured by this study are the time to complete the exercises, the number of collisions with objects in the scene, the user's feeling and the users preferable navigation methods. In addition to these information, the study also gathered the observations of the investigator and videos of the users' view through the HTC Vive. The videos were only analyzed by the investigator and helped to re-analyze the different navigation methods and the users' reactions to them.

The survey's structure is outlined in figure 4.5 to provide a quick overview. Generally speaking, the survey contained three types of tasks: Introduction tasks, practical tasks and questionnaire tasks. Introduction tasks are colored orange and include all kinds of explanations given to the user. Practical tasks, on the other hand, are colored green and comprise all tasks where the user needed to perform some kind of action with the HTC Vive or the Myo. The blue colored tasks are the questionnaire tasks where the user needed to fill out a part of the questionnaire.

At the beginning of the study, a short explanation was given to the user, covering the goals of this thesis, the study procedure, the collected data and the potential risks. The detailed explanation can be found in appendix A.2.2. As all users had German as their first language, the explanation is written in German to prevent misunderstandings due to language problems. During the explanation, the user was introduced to the HTC Vive and the Myo and was instructed to adjust both devices. Thereby, the Myo was calibrated following the instructions from the Myo Connect software.

Subsequently, the user saw a video showing a test run through the practical part. This should diminish the detriment of the first method, which may happen because the user hasn't seen the area before.

After the explanations, the user was asked to sign a consent form to confirm, that he was sufficiently informed and acknowledged that the acquired data is anonymously published in the scope of this thesis.

Subsequently, the user started with the preliminary questions of the questionnaire. Thereby, a self-generated identification code was used to map the answers of the questionnaire to the data collected during the practical part of the study. It was decided to use this identifier, as the user should be able to generate it identically again for the second user study.

Afterward, the different navigation methods were tested. Each method was first explained with the help of several illustrations showing human-like avatars wearing the Myo and performing the necessary actions. These illustrations are shown in appendix A.2.1. Subsequently, the user put on the HTC Vive and started in a test area, where he can move around as long as he needed to get used to the navigation method. By performing a double-tap gesture, the user switched from the testing area to the exercise part, which consists of a parkour-like level. The user was instructed to move to the end of the parkour efficiently, without stopping to look around. After completing the parkour, the user took off the HTC Vive and was asked to fill out the next part of the questionnaire. This procedure was repeated until the user tested all six methods.

At the end, the user was asked to complete the final part of the questionnaire and rank all methods from best to worst.

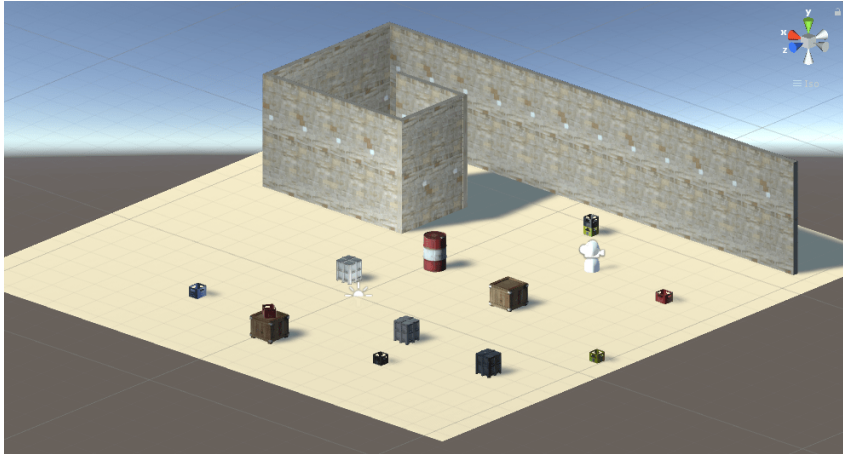


FIGURE 4.6: View onto the test area within Unity.

4.3.4 PRACTICAL PART

As already mentioned, the practical part consisted of a test area and a parkour-like level. Both levels were created in Unity using free-to-use assets from the Asset Store. The used packages were: "18 High Resolution Wall Textures" ¹, "CratesPackage" ², "Crate and Barrels" ³ and "Industrial Objects Pack" ⁴.

Test Area: The test area was inspired by the parkour and takes some of its elements like the barrels, crates, bottle-cases, and wall pieces. As depicted in figure 4.6, the elements are randomly placed in the scene to allow the user to test the control methods, navigate around objects and get a feeling for VR.

Parkour: After the user got used to the navigation method he switched to the parkour, where he started at a dead end, represented by the small white circle in figure 4.7. The parkour started with a relatively easy right curve, followed by an equally easy left curve. Afterward, the user was facing the first real exercise which is shown in detail (1) and (2) of figure 4.7. Here, the shelf shown in detail (2) was blocking the way and could only be removed by staying three seconds on the pallet, depicted in detail (1). This

¹ <https://assetstore.unity.com/packages/2d/textures-materials/brick/18-high-resolution-wall-textures-12567>

² <https://assetstore.unity.com/packages/3d/props/cratespackage-2133>

³ <https://assetstore.unity.com/packages/3d/props/industrial/crate-and-barrels-73101>

⁴ <https://assetstore.unity.com/packages/3d/environments/industrial/industrial-objects-pack-10996>

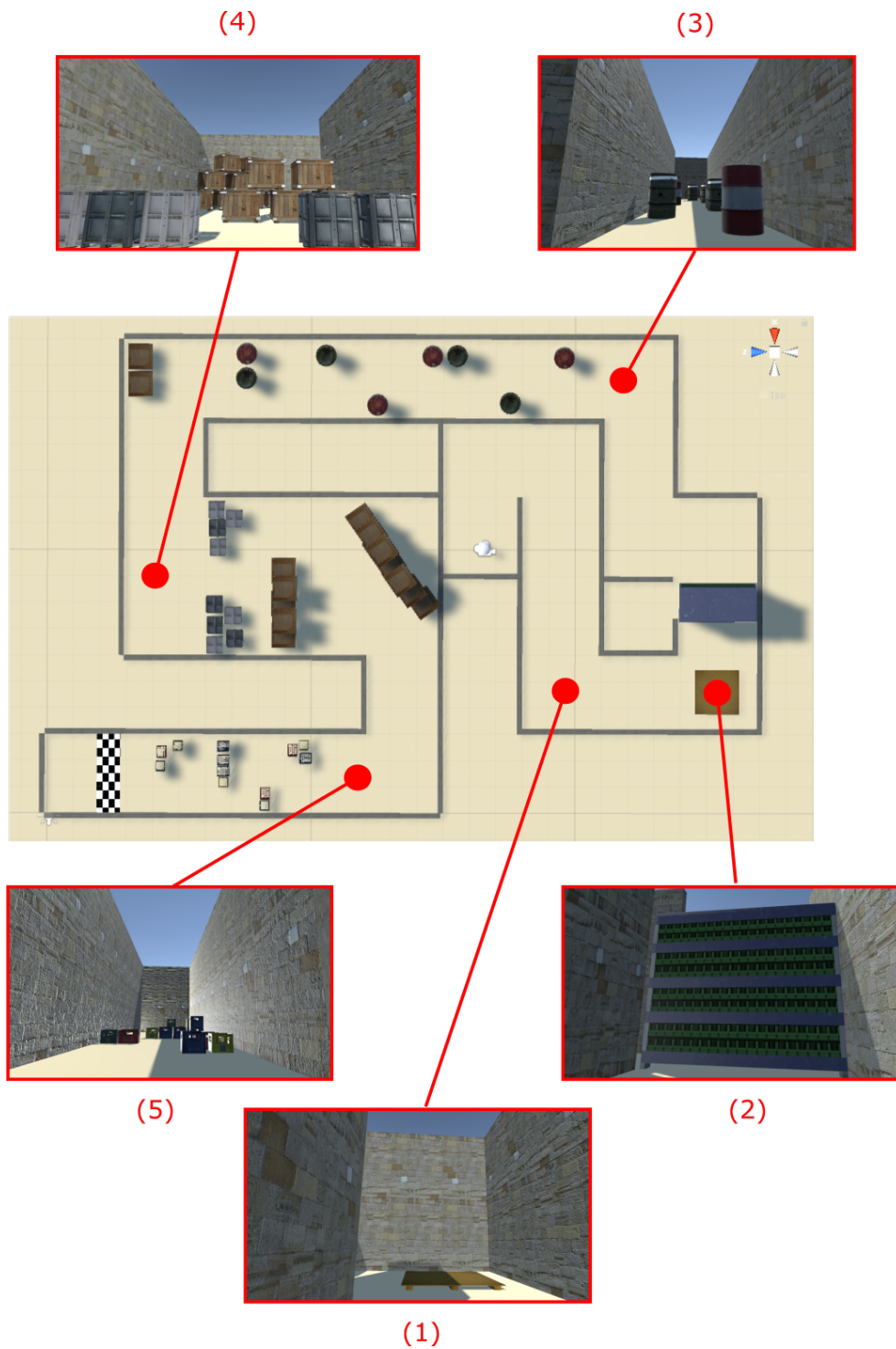


FIGURE 4.7: Top-down view on the parkour with the five most important exercises zoomed-in and highlighted in red. The white circle in the middle of the parkour represents the user and indicates the starting position. The highlighted elements are: (1) Pallet working as a switch to move the shelf. (2) Shelf. (3) Barrel-slalom. (4) Narrow passage with stacked crates. (5) Slalom made of bottle-cases.

exercise forced the user to try out the stopping mechanism of the different navigation methods and ensured, that the difficulties and problems arising during this task were also revealed.

Now, the user was able to continue his way, which lead him to the second exercise. As shown in detail (3), the user needed to move through a slalom created out of eight barrels. This task tested the handling and responsiveness of the navigation method, as the user needed to quickly and frequently change directions while moving forward.

Afterward, the user was facing the crates depicted in detail (4). These larger obstacles were meant to test precision, as the users needed to move through the small gap formed by the first row of crates. Subsequently, the user had to perform a left turn, followed by a right turn, to get through this part.

The last exercise of this parkour was the slalom made of bottle-cases shown in detail (5). As these cases were very small and close to each other, the precision, as well as the handling of the method, was tested.

When the user reached the finish line, the parkour was over and the user took off the HTC Vive to answer the corresponding part of the questionnaire.

4.3.5 QUESTIONNAIRE

The user study had a questionnaire with eight sections created with LimeSurvey.

Section A included preliminary questions asking for demographic information, main discipline, and experience with the HTC Vive and the Myo. Afterward, the user started with the practical part and returned to the survey after completing this part with the first navigation method. Section B to G were identical and asked questions about the user's experience with the tested navigation method. Thereby, questions regarding potential difficulties, the user's feelings, motion sickness, and pain were included. To gather as much feedback as possible, free text fields were added to specify information on potential difficulties, pain and general comments on the method.

After completing this section for every navigation method, the user moved on to section H where he was presented with a ranking question. This question asked the user to rank the tested navigation methods from best to worst. To help the user with distinguishing the different methods, the ranking options included the same images as the illustrations used for explaining the methods. As another form of help, the illustrations used to explain the method were numbered in the order presented to the user. The last question provided the user with the option to write down additional feedback and ideas in a free text field before submitting the answers.

For further information, a printable version of this questionnaire with all eight sections is included in appendix A.2.3.

4.4 RESULTS

This section presents the most relevant data and findings of the user study.

The first part of this chapter describes the demographic profile of the users. Afterward, the users' answers to the questionnaire are evaluated in the scope of the subjective assessment. The following objective assessment comprises the users' performance during the practical part. Subsequently, the users' additional feedback and ideas are summarized. Finally, the results are summarized to present a concise overview and conclusion. Additional results, not included in this section can be found in appendix A.2.5.

4.4.1 DEMOGRAPHIC PROFILE

The subjects of this user study were chosen from the students and staff of the TUM. Overall, 29 people took part in the study. Therefrom, seven were female and 22 were male. All subjects were between 18 and 29 years old. Three participants had already reached the master's degree. From the others, 13 participants had a high school degree and 13 a bachelor's degree.

Regarding the main discipline of the participants, five of them stated that it is not computer science. In terms of experience with VR headsets, 65.52 % had no experience and only one person had used VR headsets more than three times a week. Regarding the Myo, only three people had used it themselves before and 20 people had not even heard of it.

4.4.2 SUBJECTIVE ASSESSMENT OF THE NAVIGATION METHODS

The subjective assessment results from the evaluation of the questionnaire and is split into the users' difficulties, their assessment of their physical condition and the answers to the ranking question.

The Users' Difficulties with the Methods: According to the users' statements, the two methods with the fewest difficulties were the two "ArmPitch" methods. Thereby, the combination of the "ArmPitch" and the "ViveDirection" methods obtained the best assessments as visualized in figure 4.8.

When looking at the different types of difficulties, depicted in figure 4.9, it can be seen, that less than five people had a problem with starting and stopping a movement with the "ArmPitch" methods. In comparison to the other four navigation methods, this is

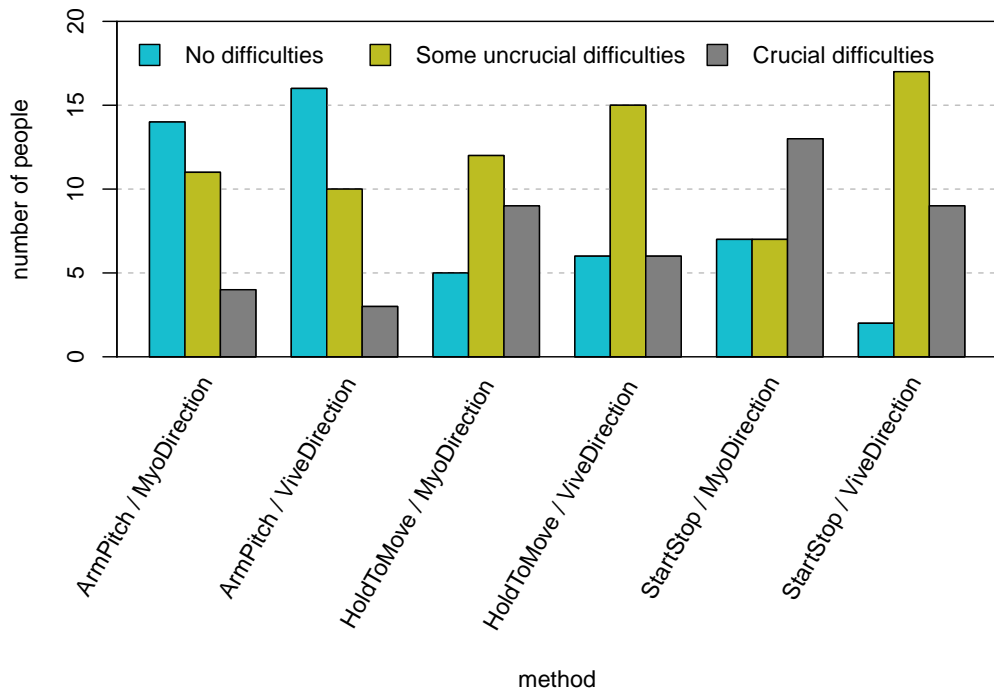


FIGURE 4.8: Bar chart showing the users' difficulties with the methods.

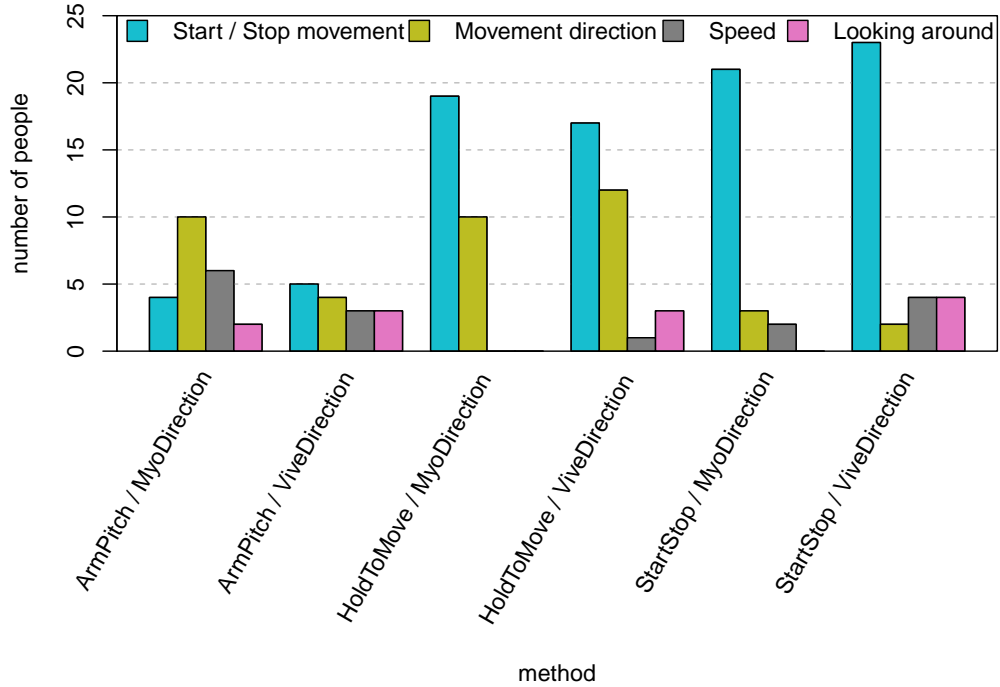


FIGURE 4.9: Bar chart depicting the users' types of difficulty.

the biggest difference, as more than half of the subjects had this kind of problem with the other methods. As this is by far the most frequently mentioned cause of difficulty, it can be assumed, that this point is accountable for the bad difficulty-assessments of the other methods.

Since some methods outperform the "ArmPitch" method regarding a few difficulties, but are not rated better overall, it can be assumed that the starting and stopping of motions is the most crucial control mechanism for users. One example is, the "Hold-ToMove & MyoDirection" method, which is better in terms of "Speed" and "Looking around", but not in the overall comparison.

The feedback and observations associated with this topic reveal, that the difficulties of starting and stopping a motion resulted from the bad gesture-recognition rate of the Myo. Users reported that the gestures were misclassified and sometimes not even recognized at all by the system. This feedback also revealed, that the problems with the "ArmPitch" method regarding this type of difficulty were because the arm needed to be held straight to stop a movement, which was especially difficult as the users were not able to see their arm.

Another interesting aspect, shown in figure 4.9, is that although the starting and stopping mechanism was identical for two methods, the number of people reporting problems with this functionality was not always the same for the two respective methods. The same observation holds for the difficulty categories "Movement direction" and "Looking around", which should yield equal results for the three methods utilizing the same `NavigationType`. This should also hold for the definition of speed, which only differs between the "ArmPitch" method and the other four. However, six people had difficulties defining the speed with the "ArmPitch & MyoDirection" method and only three with the "ArmPitch & ViveDirection" method.

Figure 4.9, also shows that less than 20.69 % of subjects had difficulties with defining the speed and looking around, which is quite promising.

The Users' Condition: In the questionnaire, the users were asked to state how much motion sickness and pain they felt during the utilization of the different navigation methods and to describe their condition in more detail if applicable.

As figure 4.10 shows, less than ten people experienced motion sickness or dizziness. The mean sickness score of those people was less than 2.5 for all navigation methods, which is still in the lower part of the scale and thus acceptable.

It is noticeable that most people felt some kind of dizziness while using the method "ArmPitch & MyoDirection", but that this is not the method with the highest sickness score. This label belongs to the "HoldToMove & MyoDirection" method, which is the

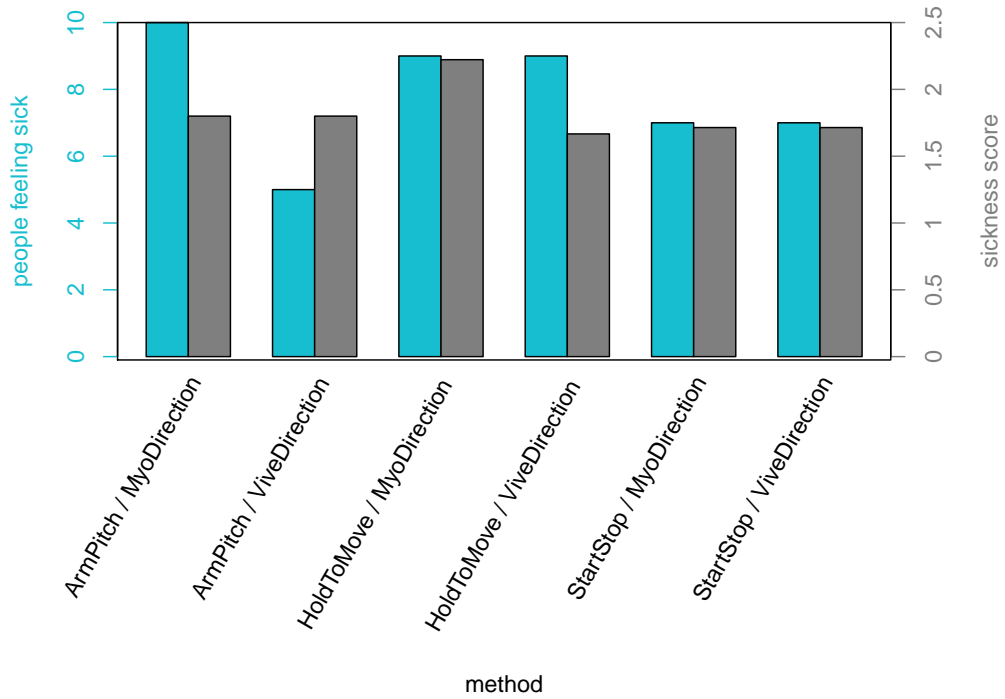


FIGURE 4.10: Bar charts visualizing the survey results regarding motion sickness. The blue bars show how many of the 29 participants experienced motion sickness and the gray bars show the mean sickness score of these people on a scale from 1 (only a little dizzy) to 5 (very dizzy).

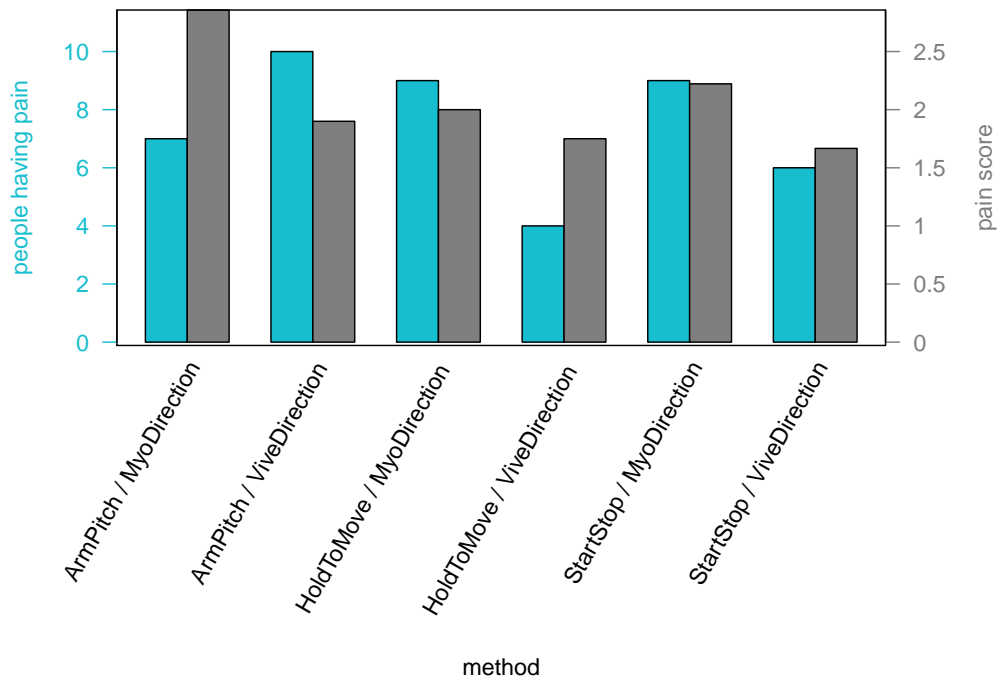


FIGURE 4.11: Bar charts visualizing the survey results regarding pain. The blue bars show how many of the 29 participants felt any pain and the gray bars show the mean pain score of these people on a scale from 1 (only a little) to 5 (very much).

only method with a sickness score above 2.0.

It can be seen that the methods using the "ViveDirection" control mechanism performed better or equal in terms of motion sickness than the methods using the "MyoDirection" control mechanism. This might be due to the independent head and body movement which allows the user to look into one direction and move into another, which might feel a little bit off and thus cause dizziness.

Besides the question regarding motion sickness, the users were also asked to assess the amount of pain they felt during the navigation tasks. Figure 4.11 illustrates, that the amount of users who specified that they felt pain is not correlated to the pain score specified by the affected users. The "ArmPitch" methods are a good example for this observation, as fewer users felt pain with the "ArmPitch & MyoDirection" method, but those users who felt pain reported a more significant pain than with the "ArmPitch & ViveDirection" method. Thus, the overall pain score of the first method is higher than the one of the second method. According to the users' feedback, this was because the first method used the arm for both, initiating a movement and defining the direction. Thus, the arm not only needed to be hold up, but also moved left and right, depending on the desired direction. Hence, the stress on the arm was higher than with the other methods which caused the arm to get tired very easily.

Regarding the pain felt during the "HoldToMove" and "StartStop" control methods, the effected users stated, that the fist gesture was tiring after some time and that the palm of their hands hurt after repeatedly performing this gesture, especially when it was not recognized correctly. In comparison to the fist gesture, only a very small amount of users stated, that they felt pain from performing the finger spread gesture.

Figure 4.11 also shows that the methods using the "ViveDirection" control mechanism performed better or equal in terms of the assessed pain score than the methods using the "MyoDirection" control mechanism.

Taking into account the users' feedback regarding their pain, it can be assumed, that the pain score can be significantly improved by abolishing the gestures and changing the mapping between speed and arm deflection.

The Rating of the Methods: In the last part of the questionnaire, the users were asked to rank all methods on a scale from best to worst. To take all ranking nuances into account the six different placement options were assigned with different values from +5 to -5. Thereby, places one to three got the positive values +5, +3 and +1 respectively assigned to them. The other placements were assigned with the corresponding negative values. Thus, a good rating increases the points of a method, and a bad rating decreases them. Hence, explicitly low ratings also effect the final result.

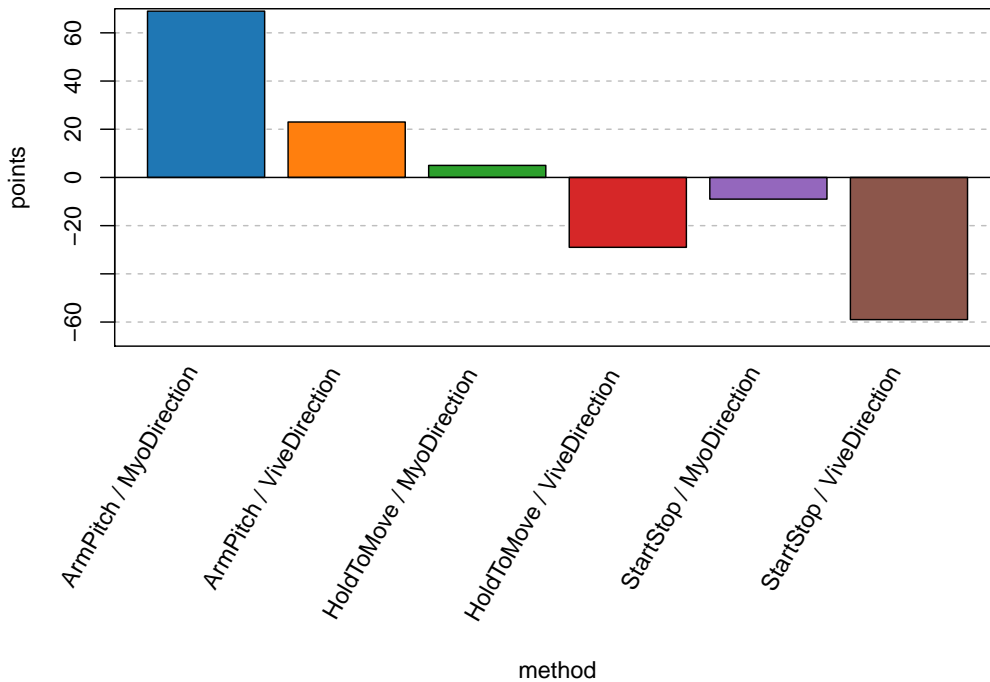


FIGURE 4.12: Point-based ranking of the different navigation methods created from the users' rankings in the questionnaire.

Figure 4.12 visualizes the final ranking resulting from the already explained point distribution on the six placement options. The resulting points show a clear placement of the methods, with the favored method being the "ArmPitch & MyoDirection" method with 69 points. With a score of 23 points, the "ArmPitch & ViveDirection" method placed second. This placement is quite unexpected as the individual results regarding difficulties, motion sickness and pain showed better results for the "ArmPitch & ViveDirection" method.

In contrast to the previously presented results on the users' condition and the difficulties with the methods, the ranking evaluation shows that in general the users prefer the "MyoDirection" control mechanism over the "ViveDirection" one.

In addition to that, it can be seen that the two methods without special gestures score considerably better in comparison to the other ones. Especially the methods utilizing the fist for starting and stopping a movement score really bad. This might be due to the fact, that the fist isn't recognized well and that this is especially problematic in cases when the users want to stop quickly. That may be also the reason why the "HoldToMove" methods scored better, as the release of a gesture is always recognized correctly and directly leads to a hold.

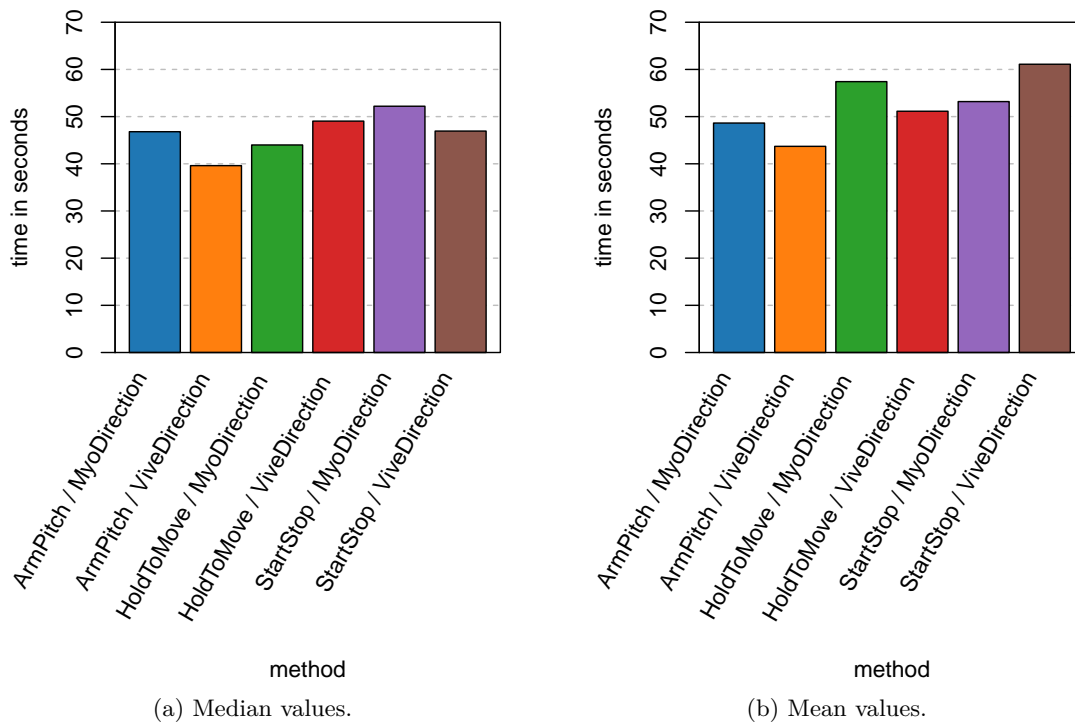


FIGURE 4.13: Visualization of two statistical measures representing the time needed to complete the parkour with the different navigation methods.

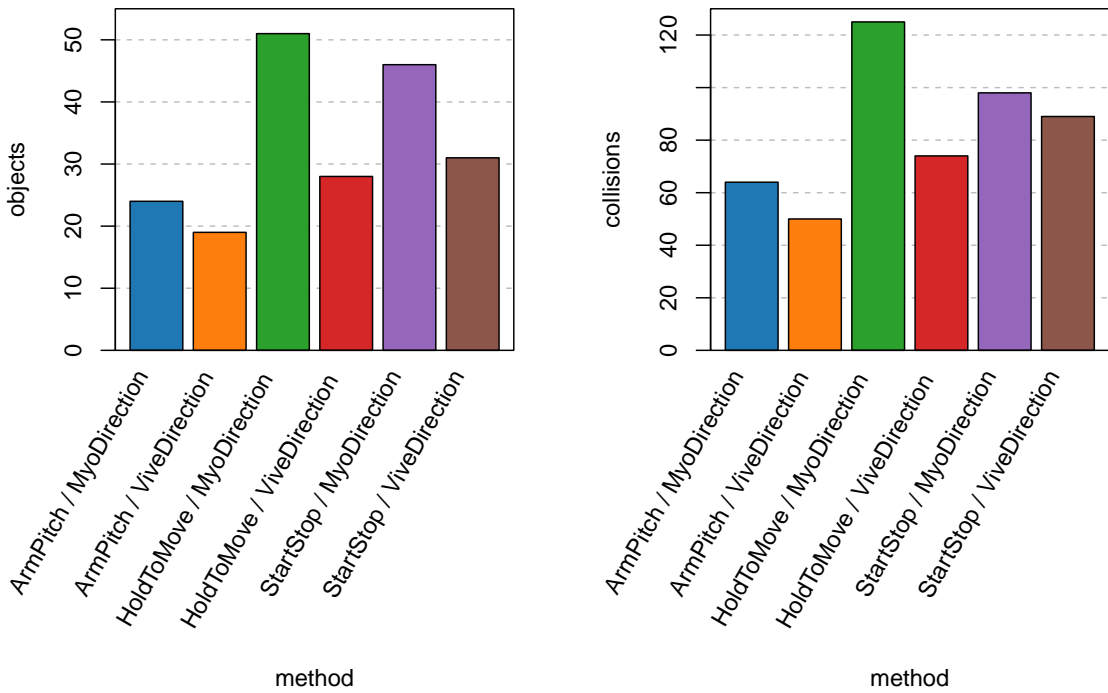
4.4.3 OBJECTIVE ASSESSMENT OF THE NAVIGATION METHODS

The objective assessment results from the data acquired during the practical part of the survey. It is split into the time needed to complete the parkour with the individual navigation methods and information regarding collisions.

Time Data: During the practical part of the survey, the time was measured from the moment the user started to move forward until he reached the finish line. The median and mean time to complete the task are shown in figure 4.13.

As a comparison of the two charts shows, the fastest method was the "ArmPitch & ViveDirection" method with 39.611 s as median and 43.698 s as mean completion time. The order of the other methods highly depends on the choice of the statistic measure. Whereas, the "StartStop & MyoDirection" method was the slowest in terms of median completion time, it was only the third slowest method in terms of mean completion time.

The "HoldToMove & MyoDirection" and the "StartStop & ViveDirection" methods, on the other hand, were considered very slow in terms of mean completion time. However, when looking at the median completion time, the first method was even considered the



(a) Amount of unique objects, all users put together hit with the respective method. Objects which were hit by more than one user are only counted once.

(b) Sum of unique collisions caused by all users together with the respective method. If a user hits an object more than once, it is only counted once.

FIGURE 4.14: Visualizations showing the number of different objects hit by all users and how often users hit objects.

second fastest.

From these results, we can conclude, that there have to be some outliers in our data. As the median is robust against outliers, this statistical measure was chosen to compare the completion times against each other. Thus, only figure 4.13a is considered from here on.

According to the median completion time, the second best method was the "HoldToMove & MyoDirection" method with 43.989s. This is 4.378s slower than the fastest method and 2.811s faster than the "ArmPitch & MyoDirection" method, which places third. Overall the results show, that users are faster when using the "ViveDirection" control mechanism. This becomes especially clear when summing up the individual median completion times of the methods using the same `NavigationType`. This results in a total median completion time of 142.989s for the "MyoDirection" control mechanism and 135.588s for the "ViveDirection" control mechanism.

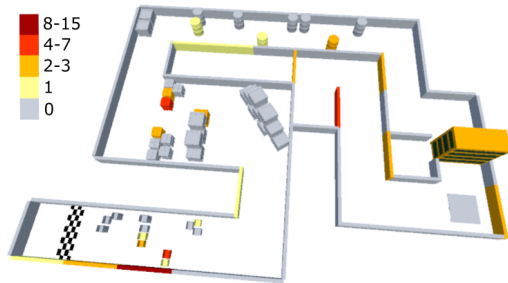
Collision Data: As a second objective measurement, the number of collisions per user and collision object was tracked for each method. This data was aggregated into the two charts depicted in figure 4.14. Chart 4.14a shows the amount of struck objects in the parkour. Which objects were hit is shown in figure 4.15. In this figure a color scheme is applied to show how many users hit which object. The more reddish the color, the more often the objects were hit. Thereby, grayed-out objects were never hit. No object was hit more than 13 times, thus, no object was hit by more than 44.83 % of all users. This is quite good as it shows, that the users are mostly able to navigate with the different methods without hitting a lot of objects.

Figure 4.14b shows the sum of all collisions per navigation method. Thereby, a collision only counts once per object and user, even if a user hit the same object multiple times. Thus, multiple hits caused by bouncing off the walls are not counted at the expense of the method.

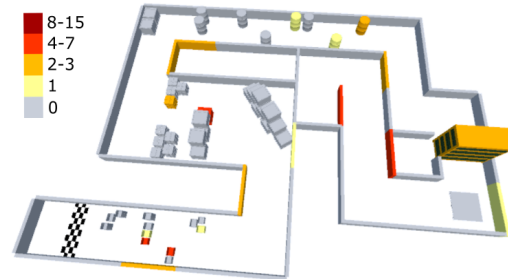
A more detailed look at figures 4.14a and 4.14b shows that the methods have the same distribution in both charts. Hence, the number of overall collisions per method is high if the number of objects struck with a method is also high. Thereby, methods utilizing the "MyoDirection" control mechanism exhibit a higher number of collisions than their counterparts. The method "HoldToMove & MyoDirection" has the worst records with 51 hit objects and 125 collisions. This result is also shown in figure 4.15c, which not only contains the highest amount of colored objects, but also the highest amount of red objects. The high number of colored objects shown in figure 4.15e coincides with the data in chart 4.14a and 4.14b, showing that the "StartStop & MyoDirection" method exhibits the second most collisions with 46 hit objects and 98 collisions.

In contrast to the other "MyoDirection" methods, the "ArmPitch & MyoDirection" method performs much better as it exhibits the second least collisions with 24 hit objects and 64 collisions. The only method which performs even better is the "ArmPitch & ViveDirection" method with 19 hit objects and 50 collisions.

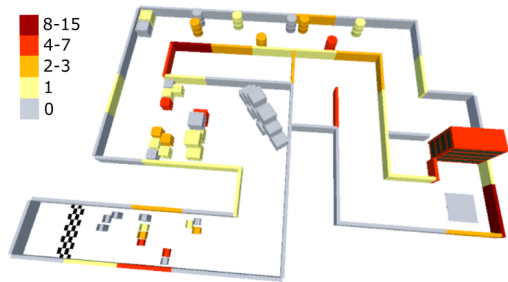
Figure 4.15 shows that most users had problems with stopping their movements on the pallet to move the shelf and thus ran into the wall behind it. Especially the methods utilizing the "HoldToMove" and "StartStop" control mechanism show a high amount of collisions, which implies that it is difficult to stop with these methods. Another interesting detail shown by these collision maps is, that narrow passages and turns like the turn at the end of the barrel slalom or the passage in the middle with the stacked crates and the bottle-case slalom at the end impose a high challenge to the users. These three parts show collisions in nearly all collision maps and show especially bad results for the "HoldToMove" and "StartStop" methods, implying that it is difficult to precisely and quickly navigate with these navigation techniques.



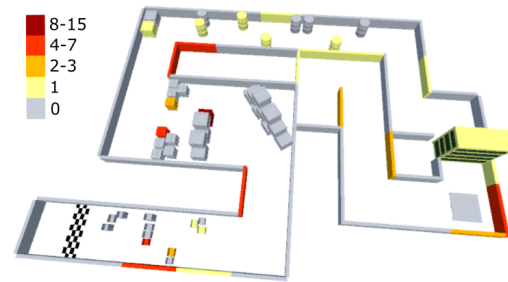
(a) Collision-map of the "ArmPitch & MyoDirection" method.



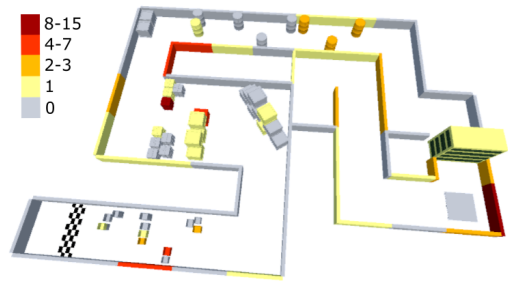
(b) Collision-map of the "ArmPitch & ViveDirection" method.



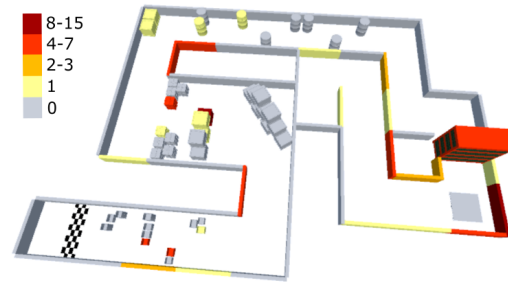
(c) Collision-map of the "HoldToMove & MyoDirection" method.



(d) Collision-map of the "HoldToMove & ViveDirection" method.



(e) Collision-map of the "StartStop & MyoDirection" method.



(f) Collision-map of the "StartStop & ViveDirection" method.

FIGURE 4.15: Visualization of the users' collisions with the different navigation methods. The color indicates how many different users collided with an object while using the specified navigation method.

4.4.4 AGGREGATED FEEDBACK

The users were given several opportunities to write down feedback. This feedback is categorized into the sections: "Movement Mode", "Movement Direction", "Pain" and "Overall Feedback". As the feedback regarding "Pain" was already discussed in section 4.4.2 under the term "Users' Condition", only the other three categories are presented here.

Feedback to the Movement Mode: Regarding the "ArmPitch" control mechanism, the users stated, that they accidentally put down the arm to stop when suddenly hitting an object or quickly wanted to stop. As this behavior did not lead to the desired stop, but to a backward movement instead, it caused some confusion and frustration.

To solve this problem, one user suggested, to stop the movement for the "ArmPitch" methods by putting the arm down and accelerate by lifting it up until it reaches a horizontal position. Another user made a similar suggestion, stating that the range of speed distribution should be changed so that the arm doesn't have to be raised up that high. This would also solve the problem of pain in the arm and shoulder, due to constantly holding up the arm.

The most frequent feedback regarding the "HoldToMove" methods was, that the fist gesture was not recognized well which caused severe problems with the movements. Another feedback regarding the "HoldToMove" and the "ArmPitch" methods was, that the backward movement seemed very useless as the users were not able to look backward. One user suggested to change the backward movement to have slow, fixed speed and fixed direction to be able to easily navigate out of objects.

The feedback for the "StartStop" methods revealed, that users had problems with starting and stopping the movement, as the fist was recognized poorly. One user said, that the lack of backward movement was stressing. It was also stated by one user, that this form of walking was easier as no active task was needed to be performed to keep the avatar moving.

Feedback Regarding the Movement Direction: Generally speaking, the users are in disagreement on the preferred direction control method. A lot of users stated, that they prefer a method where viewing and moving direction are separated from each other ("MyoDirection"), however, others stated, that the independent head movement caused motion sickness and that the navigation with the "MyoDirection" methods was more complicated as they could not see their arm. A significant amount of users also said that they prefer the "ViveDirection" methods, as they were easier to use and more immersive. However, others said that looking around with these methods was very

difficult. In conclusion, there is no optimal direction control method as both methods have their advantages and disadvantages. One user summed it up very well by saying, that the "ViveDirection" method may be easier to learn and use, but the "MyoDirection" method provides more possibilities.

Overall Feedback and Ideas: Most users stated, that the "ArmPitch" methods were overall good, easy to use and intuitive.

Regarding the other methods, the users noted, that an improved gesture-recognition would be beneficial. For the "ArmPitch" methods, the users do not agree with each other, as some said that they are not intuitive and others stated that they were intuitive. One person even said that they would be his favorite methods if he would not need to use the fist gesture.

Similar comments can be found in relation to the "StartStop" methods, where some users think that these methods are very fluid and intuitive and others stated that they are very bad and that they felt uncomfortable while using them.

Summing up the feedback it can be concluded, that the speed mapping needs to be changed so that the arm does not have to be lifted so high and that the gestures must be substituted or removed from the methods. Regarding the direction control method, no universal valid answer could be found as the assessments of the users were very diverse and contrary. However, multiple users mentioned that they found the methods and especially the speed control intuitive, which is an indication that the methods go into the right direction.

4.4.5 SUMMARY AND IMPLICATIONS

It can be said, that the "ArmPitch" methods performed better in every category than the other methods as they didn't utilize the gesture recognition provided by the Myo. Thus, a good navigation method needs to either adopt the "ArmPitch & MyoDirection" method or the "ArmPitch & ViveDirection" method. The subjective assessment showed, that the users reported more difficulties with the "MyoDirection" control mechanism. However, fewer users experienced pain with this method and this might also be the reason why their subjective rating of the methods, ranked the "MyoDirection" control mechanism at first place. However, the objective assessment showed, that the users were not only faster but also caused lesser collisions while utilizing the "ViveDirection" control mechanism. Even though, the users performed better with the "ViveDirection" control mechanism, they still prefer the "MyoDirection" control method. The users' feedback showed, that they indeed not agree on the question which control mechanism should be used, as they stated, that the "ViveDirection" method is easier to learn, but

the "MyoDirection" method provides more possibilities. This just reflects the discrepancy between the subjective and objective assessment as well.

Taking everything into consideration, it was decided to further develop the "ArmPitch & MyoDirection" method. This decision was made as this method does not only provide more navigation possibilities and was rated higher by the users, but also as it provides the freedom to look around while moving. The last point is especially important in the context of the NRP, where researchers should be able to observe the robot's behavior while moving.

To improve this method it was decided to adopt the feedback regarding the arm pitch mapping for starting and stopping the movement. Moreover, it was decided to adopt the suggestion of a backward movement with fixed speed and fixed direction to make this movement easier and more predictable.

CHAPTER 5

COMPARISON OF GESTURE- AND CONTROLLER-BASED NAVIGATION

After the first user study found a navigation method with sufficient potential, this method was refined according to the users' feedback and implemented in the Unity3D-Client. To analyze the overall potential of the refined method, a controller-based navigation approach was implemented as a reference for the evaluation.

This chapter presents the whole procedure from the refinement and implementation of the gesture-based navigation method, over the adaptations on the Unity3D-Client and the implementation of the controller-based method, to the final user study.

Thereby, section 5.1 covers the refinements of the gesture-based navigation method and the development of the controller-based one.

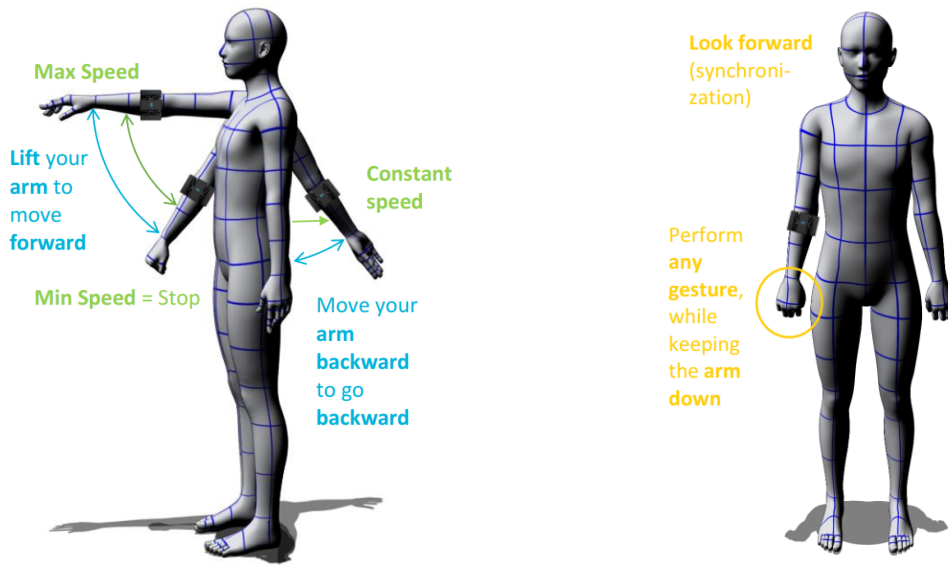
Section 5.2 describes the changes which were done to the Unity3D-Client and the implementation of the two navigation approaches. It also includes a listing of the utilized tools and technologies.

The evaluation method, its goals, and the setup are shown in section 5.3.

Finally the results of the evaluation are summarized in section 5.4, which not only includes the objective, but also the subjective assessments of the users and their feedback.

5.1 APPROACH

For a qualitative comparison of the selected gesture-based navigation method against a controller-based one, the selected method needed to be refined according to the users'



(a) Illustration of the refined arm-pitch mapping.

(b) Explanation on how to toggle the two modes.

FIGURE 5.1: Visualizations showing how the refined gesture-based navigation method works.

suggestions. In addition to that, the controller-based navigation method needed to be designed in accordance with established controller-based navigation standards.

5.1.1 REFINEMENT OF THE GESTURE-BASED NAVIGATION METHOD

As already explained in section 4.4.5, the method "ArmPitch & MyoDirection" was chosen to be refined and reimplemented in the Unity3D-Client.

Thereby, it was decided to include the following refinements.

The problem of pain in the shoulder due to constantly holding up the arm was solved by changing the maximal pitch of the users' arm. Now, the maximal speed is reached if the arm is horizontally outstretched in front of the user as depicted in figure 5.1a.

Following the feedback acquired during the first user study, the movement stops if the arm is put straight down. In addition to that, a backward movement with constant speed and direction was included in the refined approach. This motion is initiated by putting the arm backward as shown in figure 5.1a.

In contrast to the control mechanism for starting and stopping the movement, the direction control method stays the same: The user moves into the direction the Myo is pointing at and is able to freely look around through the head-tracking technology of the HTC Vive.

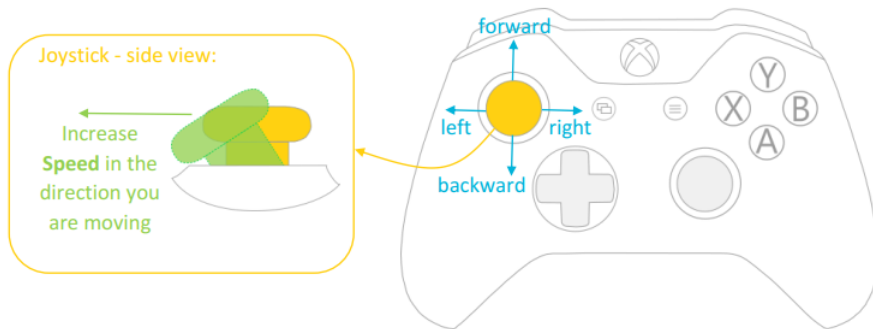


FIGURE 5.2: Visualization of the controller-based navigation method.

Since the avatar's movements are only a part of the action set that should be performed, enough interaction possibilities needed to be kept open to easily implement further actions in the future. Hence, two modes were included to separate the avatar's movements from other interactions. This allows to use the same set of gestures and motions for the navigation method and the interaction control.

To toggle the modes, a gesture or motion was needed. As the users clearly stated, that the gesture-recognition capabilities of the Myo are not good enough to be used as a central part of the navigation, it was decided to build a trigger from the raw EMG data. The final trigger requires the user to put his arm down, next to his body, as depicted in figure 5.1b, and perform any hand gesture, not necessarily one of the predefined poses. Thereby, it is important that the user looks forward, as this moment is used to synchronize the coordinate system of the Myo with the one of the HTC Vive. We chose this type of synchronization and not a distinct synchronization phase, as the users should be able to freely look around and kneel down if not in movement mode.

During this phase, the HTC Vive is decoupled from the corresponding Unity object, which causes the two coordinate frames to diverge. Hence, the deviation needs to be corrected, whenever the users enter the movement mode.

5.1.2 DESIGN OF THE CONTROLLER-BASED NAVIGATION METHOD

For a qualitative good evaluation of the gesture-based navigation approach, the controller-based method needed to be designed according to established standards. Thus, an Xbox controller was used, as this is one of the most widely used controllers right now. Thereby, the left joystick is used to control the user's movements, just as in many computer games. The control of the users' speed was also adopted from computer games and is defined through the deflection of the left joystick as depicted in figure 5.2.

In contrast to many computer games, the user’s rotation is not controlled through the right joystick, but rather through the physical rotations of the user measured by the HTC Vive. The reason for this decision was the research of Riecke et al. [Rie10], where they found out, that a combination of joystick navigation with physical rotations might have significant benefits over mere joystick navigation. McCullough et al. [McC15] also used this combination to evaluate their gesture-based navigation method.

5.2 IMPLEMENTATION

For the implementation of the two navigation methods, the tools and technologies described in section 5.2.1 were used. The adaptations of the Unity3D-Client, as well as the implementation of the avatar’s rotation and the two methods were done with *C#* and are described in sections 5.2.2, 5.2.3, 5.2.4 and 5.2.5 respectively.

5.2.1 UTILIZED TOOLS AND TECHNOLOGIES

This section provides a short overview of the utilized tools and technologies. A detailed description can be found in table 5.1. As most of the technologies were already described in section 4.2.1, this section only includes descriptions of the newly added technologies.

NRP Server: To have full backend and frontend access, the NRP was installed locally on Ubuntu 16.04. The machine was located in the chair of Human-Machine Communication at the TUM just a few doors away from the computer used for the evaluation. Thus, the network delay between the computer and the server was negligible.

MyoUnityRawEmg: This GitHub gist from Chris Zaharia ¹ is a patch for the Myo SDK which provides access to the raw EMG data of the Myo.

5.2.2 CHANGES ON THE UNITY3D-CLIENT

As already mentioned in section 2.1.4, the Unity3D-Client was used as a basis for the implementation and deals with the connection to the NRP and the topic-based communication. The original client already includes the rendering and updating of the scene

¹<https://gist.github.com/chrisjz/efb6d3aa53fd65fb2364>

Computer Specifications	
Processor	Intel(R) Core(TM) i7-6700K
RAM	16 GB
OS	64-bit Windows 10 Education
Graphics Card	NVIDIA GeForce GTX 970
NRP Server Specifications	
Processor	Intel(R) Core(TM) i5
RAM	4 GB
OS	Ubuntu 16.04.3 LTS
Graphics Card	NVIDIA GeForce GTS 250
Unity	
Version	5.6.2f1
Microsoft Visual Studio Community 2017	
Version	15.4.5
HTC Vive Headset	
Screen	Dual AMOLED 3.6" diagonal
Resolution	1080x1200 pixels per eye
Refresh rate	90 Hz
Field of view	110 degrees
SteamVR	
Version	1515522829
SteamVR Plugin for Unity	
Version	1.2.2
Myo	
Firmware Version	1.5.1970
Model	MYOD5
Part Nr.	MYO-00002-001
Myo Connect	
Version	1.0.1
Myo SDK	
Version	0.9.0 for Windows
MyoUnityRawEmg	
Last update	24.12.2014
Supported Myo SDK version	0.8.0 for Windows
LimeSurvey	
Version	2.72.5+171121

TABLE 5.1: Utilized tools and technologies with the appropriate versions and informations.

simulated at the NRP. Therefore, the client subscribes to the Gazebo topics `~/scene`, `~/pose/info`, `~/model/info` and `~/material`. A detailed sequence diagram showing all communication steps between client and server can be found in appendix A.1.

Although the client includes the scripts required for a connection to the ROS bridge, no appropriate subscriber or publisher scripts were included yet. Those scripts encapsulate the message's topic, its content, its format and a callback function if needed. Fortunately, the avatar object in the NRP already provides topics to get and set its rotation and velocity. Thus, only the corresponding publisher scripts `ROSAvatarRotPublisher` and `ROSAvatarVelPublisher` needed to be implemented in the Unity3D-Client. Therefore, the appropriate base classes provided by the `ROSBridgeLibrary` were used. The two publishers utilize the predefined message types `geometry_msgs/Quaternion` and `geometry_msgs/Vector3` respectively.

Given that the avatar is already instantiated on the NRP, the client is able to utilize the publishers to transfer movement data to the avatar. If the avatar has not been instantiated yet, publishing the movement data does not lead to a motion.

Since the simulations on the NRP usually do not contain an avatar, the user would have to manually instantiate one, every time he starts a new simulation. Thus, we decided to adapt the Unity3D-Client to instantiate an avatar if not already present in the scene. Therefore, the initial rendering of the scene, coming from the topic `~/scene`, is used to search for a `GameObject` which name contains the word "user_avatar". If no such object is found, the avatar is created by sending a corresponding message to the topic `~/factor`. This message contains the model's name, the SDF file name of the desired model and its coordinates.

To allow multiple different users inside the simulation simultaneously, the avatar's name consists of the prefix "user_avatar_" and the user's IP address, whereas, the dots in the IP address are replaced with underscores.

5.2.3 IMPLEMENTATION OF THE PHYSICAL AVATAR-ROTATION

As already stated in section 5.1, the avatar's rotation should be controlled through the user's physical rotation. Therefore, the script `AvatarRotations` was added to the `GameObject Camera (head)`, which is a child-object of `[CameraRig]`. This script publishes the rotation of the HTC Vive over the corresponding ROS publishers to the server.

In addition to that, the script `VRMountToAvatarHeadset` was added to the `GameObject [CameraRig]` to prevent the user from moving their view without also moving the

avatar. Thereby, the position changes of the HTC Vive are compensated by adapting the position of [CameraRig] accordingly.

However, this adaption comes with the disadvantage, that the user is not able to use the full capabilities of the HTC Vive anymore. This includes, crouching down, which might be needed in the future to take a closer look at objects and interact with them more easily. Thus, the compensation of the HTC Vive's position changes is only activated when the user wants to move. This happens for the controller-based navigation method when the user pushes the joystick and for the gesture-based method when the user is currently in the movement mode.

5.2.4 IMPLEMENTATION OF THE GESTURE-BASED NAVIGATION METHOD

The gesture-based navigation method was implemented similar to the "ArmPitch" method and is structured into three logical sections.

The first section, the synchronization part, aligns the coordinate frames of the Myo and the HTC Vive. Therefore, an anti-yaw and an anti-roll are added to the forward vector of the Myo.

The second part allows the user to toggle between the movement and interaction mode. Therefore, the user needs to put down the arm, next to his body and perform any hand-motion. The first step in recognizing the toggle-motion is to detect if the user's arm is pointing down and located next to his body. Therefore, the y-value of the Myo's forward-vector needs to be smaller than an empirically identified deflection minimum of -0.8. In addition to that, the relative roll of the user's arm needs to be smaller than 20, which ensures, that the arm is pointing downward and not backward or sideward instead. If the user's arm is in the correct position, the raw EMG data generated by the eight EMG sensors of the Myo is used to detect any hand-motion. Thereby, a motion is recognized if one of these sensors transmits a value larger than the empirically discovered threshold value of 85. Even if all these prerequisites are fulfilled, the user is only able to switch into the movement mode if the y-value of the HTC Vive is larger than 75 % of the user's original height. Thus, the user can not accidentally switch into the movement mode while crouching down, which may cause an unexpected movement which in turn would lead to motion sickness.

The last section deals with the control of the actual movement, which is only activated during the movement mode. A movement is initiated by moving the arm upwards, which causes the y-value of the Myo's forward-vector to get larger than the minimal

deflection value of -0.8. A corresponding backward movement is initiated by a y-value smaller or equal to the minimal deflection value and a relative roll greater than or equal to 90, which is only the case if the arm is pointing backward.

Depending on the desired movement direction, the speed is either defined through the y-value or set to a constant speed respectively. In both cases the forward-vector of the Myo is multiplied with the speed and published to the corresponding ROS topic as movement velocity. Thereby, the y- and z-value of the vector are switched as the coordinate frame of the NRP is rotated compared to Unity's coordinate frame.

During the movement, a coroutine is used to trigger a pulsing-vibration in the Myo whenever the player is moving.

5.2.5 IMPLEMENTATION OF THE CONTROLLER-BASED NAVIGATION METHOD

The controller-based navigation method is much simpler in terms of implementation than the gesture-based one. If the joystick's deflection is larger than a predefined threshold, its x-axis is taken as x-value, and its y-axis as z-value for the movement's direction. To take the user's rotation into account as well, the direction vector is multiplied with the avatar's rotation.

The speed is calculated based on the maximal deflection of the joystick and a linear function with predefined values. Thereby, either the absolute deflection of the joystick's x- or y-axis is used, depending on which value is larger. Afterward, the speed is multiplied with the direction-vector and published to the corresponding ROS topic as movement velocity. Since the coordinate frame of the NRP is different from the one in Unity, the y- and z-value are switched when publishing the velocity to the server.

5.3 EVALUATION

This section summarizes the goals, setup and structure of the final user study, which is named: "User Study to Compare Gesture- and Controller-Based Navigation in VR".

5.3.1 GOALS

One of the evaluation's goals was to find out if the gesture-based navigation method could be improved in comparison to the previously tested methods. Another goal was to evaluate how the gesture-based approach performs in comparison to a controller-based method, which most users already used before.

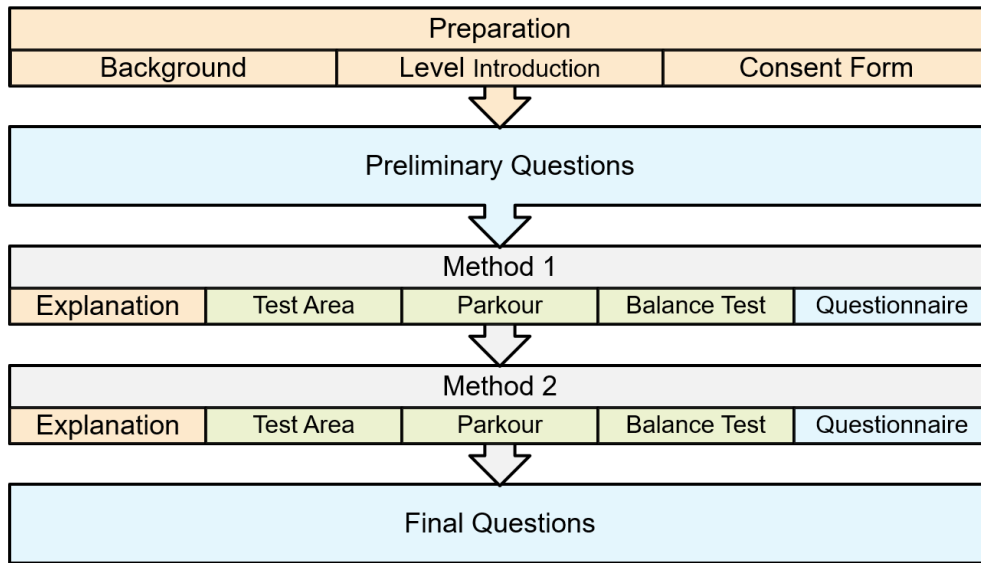


FIGURE 5.3: A schematic overview of the evaluation procedure.

5.3.2 SETUP AND STRUCTURE

During the evaluation the technologies and tools mentioned in section 5.2.1 were used. Like the last study, this one again took place in the Augmented-/Virtual-Reality Lab of the chair of Human-Machine Communication at the TUM. The exercises of the practical part took place on a dedicated HTC Vive computer and the questionnaire was shown on a separate laptop.

Following the same procedure as the last user study, this one consisted of a practical part, a questionnaire and the observations of the investigator. This user study also followed a repeated measures design due to the same reason of fewer required users. To prevent a bias caused by the order in which the users tested the two methods, the order was randomly chosen for each user. The only restriction to this random selection was that an equal amount of users from the first user study needed to start with the gesture-based navigation method and with the controller-based one.

The independent variables of this evaluation are the two control methods. Whereas the dependent variables are the time to complete the practical part, the users' feelings, the preferred navigation method and the number of times, the users fall down from the runway in the last level of the practical part. In addition to this information, the evaluation also gathers videos of the users' view through the HTC Vive. The videos were only analyzed by the investigator and helped to reanalyze the different navigation methods and the users' reactions to them.

An overview of the survey's structure is outlined in figure 5.3. Following the structure of the last user study, this one contains three types of tasks: Introduction tasks, practical tasks and questionnaire tasks. Introduction tasks are colored orange and include all kinds of explanations given to the user. Practical tasks, on the other hand, are colored green and comprise all tasks where the user needed to perform some kind of action with the HTC Vive, the Myo or the controller. The blue colored tasks are the questionnaire tasks where the user needed to fill out one part of the questionnaire.

Since also new users took part in this study, the goals of this thesis were explained again at the beginning. Afterward, the study procedure, the collected data, and the potential risks were explained to the user. Since all users had German as their first language, the explanations, which can be found in appendix A.3.2 were given solely in German. During the explanation, the new users were introduced to the HTC Vive and the Myo. In contrast to the last user study, the Myo was not calibrated as its gesture recognition capabilities were not used. After the introduction, the user saw three videos presenting the three different levels of the practical part. This approach should diminish the detriment of the first method, which may be caused due to the fact, that the user has not seen the area before. After the whole introduction, the user was asked to sign a consent form to confirm, that he were sufficiently informed and acknowledged that the acquired data is anonymously published in the scope of this thesis.

Subsequently, the user started with the first part of the questionnaire, which contained preliminary questions covering the previous knowledge with the tools. Thereby, the same self-generated identification code as for the first user study was used. Thus, users which already took part in the first user study could be easily identified.

Thereafter, the two navigation methods were tested by completing three different levels. The first level was a test area, which allowed the user to get used to the navigation method. For the following two levels, the user was instructed to complete them efficiently, without stopping to look around or do other things. Before the user started with the levels, the corresponding navigation method was explained with the help of several illustrations included in appendix A.3.1.

As the levels were simulated on the NRP, they needed to be switched on the server whenever the user finished a level. Since this caused small delays, the user was instructed to keep the HTC Vive headset on and wait until the next level starts. Thereby, the begin of the next level was introduced by the observer through a question asking the user if he was ready.

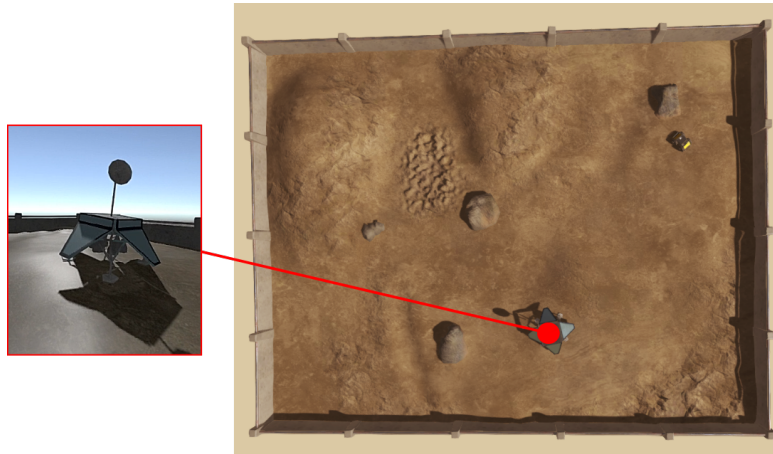


FIGURE 5.4: View onto the test area within the NRP.

After the user completed all three levels, he were asked to fill out the corresponding part of the questionnaire. This procedure was repeated for the second navigation method.

Thereafter, the user was asked to complete the last part of the questionnaire and choose his favorite method. In addition to that, the user was asked to explain why he favored this method over the other.

5.3.3 PRACTICAL PART

The practical part consisted of three consecutive levels. The first level was designed to be a playground to get used to the navigation method. The other two levels should test how well and precise the user is able to navigate with the method.

All levels were designed using the `Environment Editor` of the NRP.

Test Area: For the test area, the SpaceBotCup 2013 Arena from the template files was used. This area provides enough space for the user to move around and also contains hills and stones which show the user how moving upward in VR feels like.

In addition to that, the area also contains the Husky robot which slowly moves around in circles next to the stone, which is shown in the top right of figure 5.4. Thus, the user was able to inspect the robot from close up, which was not possible in the other two levels.

The blue satellite highlighted in figure 5.4 served as an endpoint for this level. The user was instructed to move into this satellite when he thought, that he knew how to navigate with the method.

Parkour: After the user adjusted to the navigation method, the level was switched to the parkour where the user started in front of a large glass door as shown in figure 5.5. Thus, the user was only able to move in one direction.

The parkour started with a relatively easy slalom through office chairs as shown in detail (1). This slalom was created to test the handling of the navigation method, as the user needed to frequently change directions while moving forward.

At the end of this slalom, the user came to a glass front, where he needed to make a left turn and move past a padded chair as detail (2) shows. The chair was placed in a way, that the user could not see it from afar, which forced him to spontaneously react to this obstacle. Thereby, the responsiveness of the navigation method should be tested.

As shown in detail (3), the next exercise was a narrow slalom made out of dustbins. This slalom was smaller than the first one and led to a dead end formed by the wall and the couch.

From there on the user needed to navigate through the small gap shown in detail (4) to move on to the Pioneer robot depicted in detail (5). When the user reached the robot and moved into it, the level was finished and the next level was started.

Balance Test: The last level was used to test how precise the user is able to navigate with the two methods. From the slalom part, the user first saw the cage depicted in detail (1) which was made out of glass and contained the robot which formed the target of this level.

To reach the robot the user had to move up the stairs shown in detail (2) to get on the small runway depicted in detail (3). Then, the user needed to follow the runway to get to the back of the glass cage.

The runaway contained a few narrow turns which require high movement precision. The turns forced the users to move slower, which tested the speed control of the two methods. If the user fell down, he had to go back to the stairs and start again.

When the user reached the back of the glass cage, he needed to move through the gap depicted in detail (4) to get to the robot. To finish the level, the user had to walk into the robot shown in detail (5).

As this was the last level, the users were instructed to take off the HTC Vive and fill out the corresponding part of the questionnaire.

5.3.4 QUESTIONNAIRE

To gather the user's assessments of the methods, a questionnaire consisting of four sections was created with LimeSurvey.

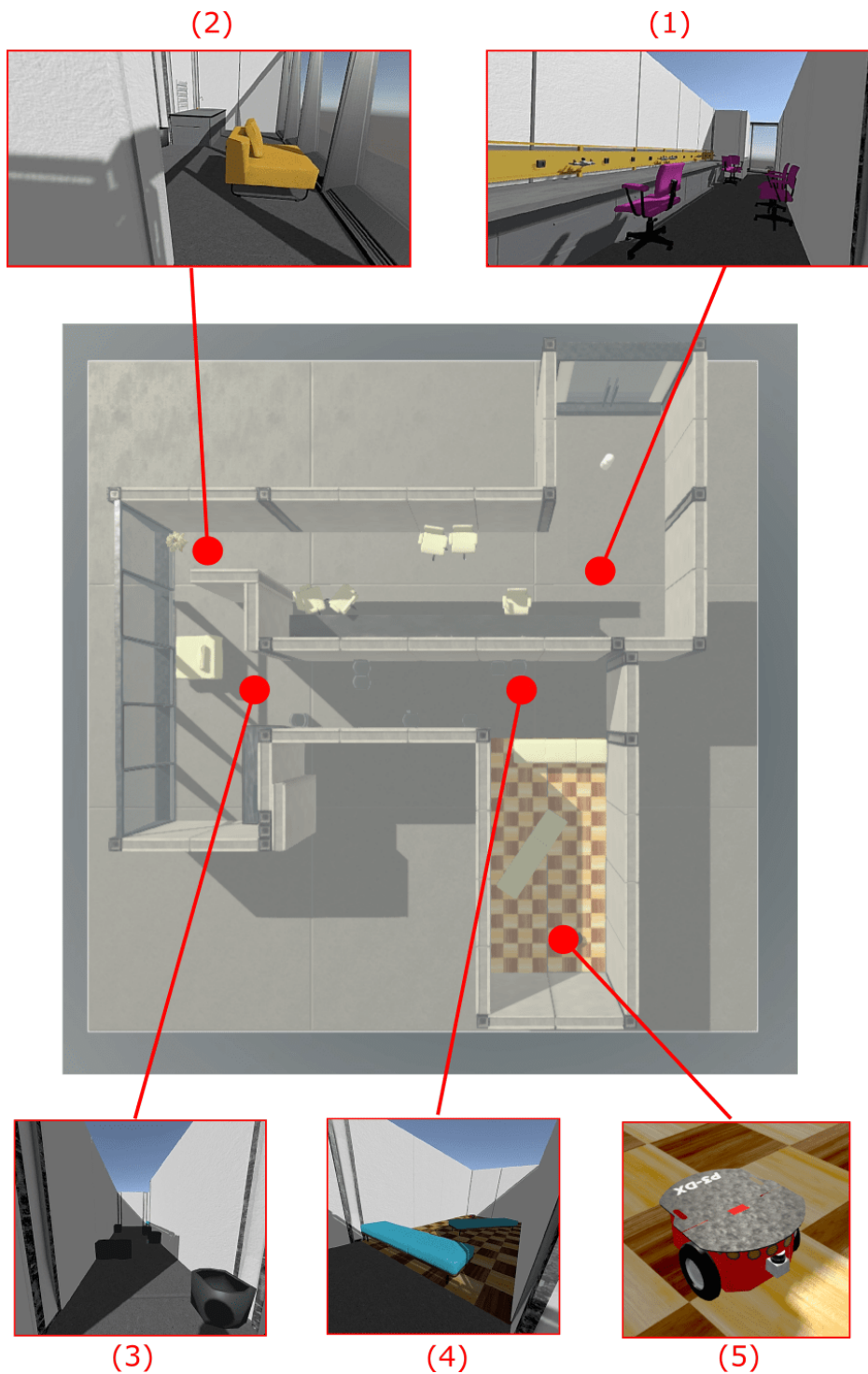


FIGURE 5.5: Top-down view on the parkour with the five most important exercises zoomed-in and highlighted in red. The user starts in front of the glass door on the top right. The highlighted elements are: (1) Slalom made of office chairs. (2) Passage created with the wall and the padded chair. (3) Smaller slalom made of dustbins. (4) A narrow passage between wall and couch. (5) Pioneer robot as target.

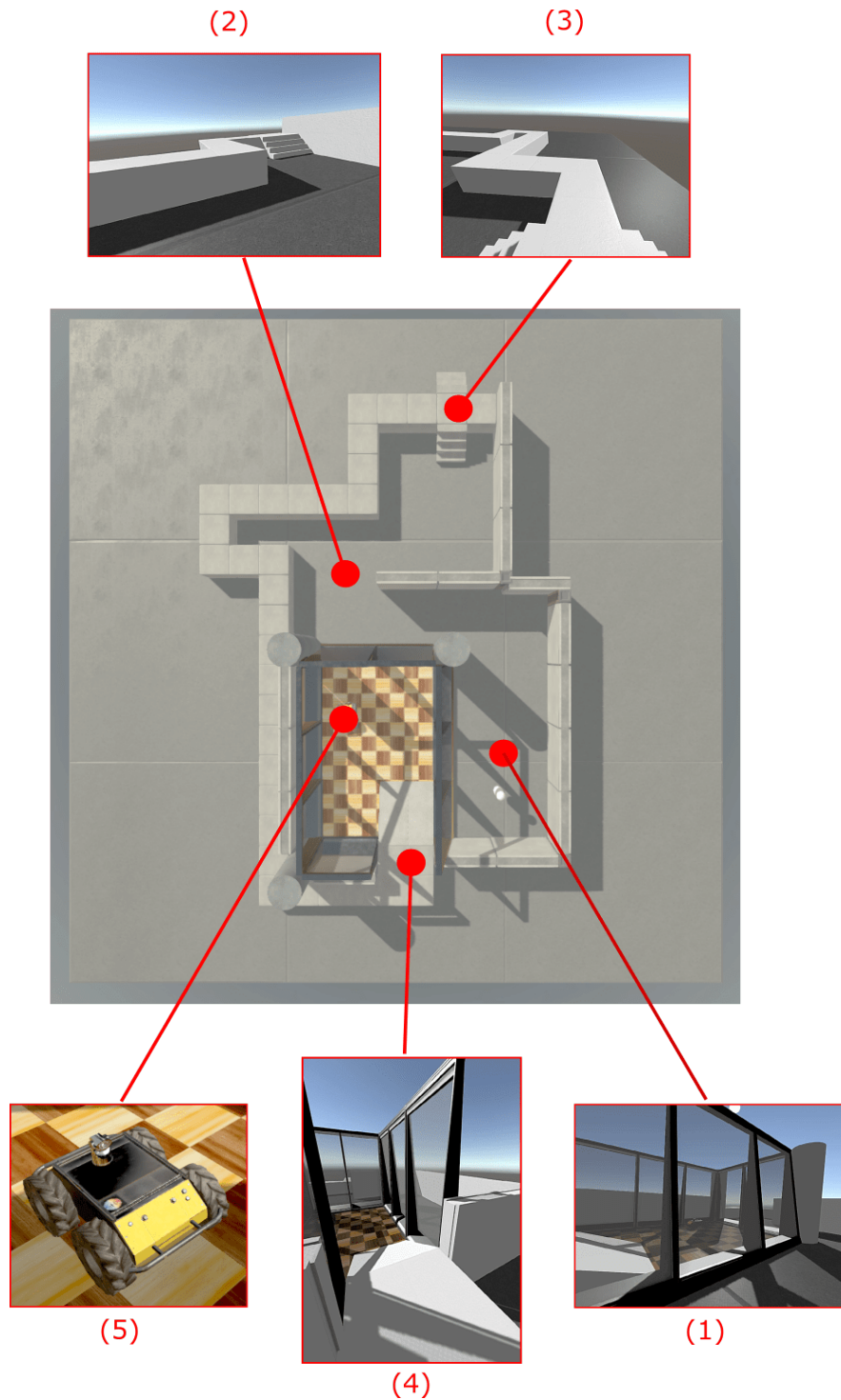


FIGURE 5.6: Top-down view on the balance test level with the five most important exercises zoomed-in and highlighted in red. The user starts on the bottom right of the level. The highlighted elements are: (1) View on the glass cage trapping the robot. (2) Stairs which the user needs to use to get onto the small runway. (3) View onto the small runway which the user needs to balance on to get to the robot. (4) A gap in the glass cage which allows the user to get down to the robot. (5) Husky robot as target.

Section one only contained preliminary questions, which asked for demographic information, main discipline, experience with the HTC Vive and the Myo, and usage of controllers for playing video games.

Section two and three asked for potential difficulties with the tested navigation method, the user's feelings, and his condition while using the method. To gather as many information as possible, free text fields were added to the questionnaire, to allow the user to specify the kind of difficulties and pain he experienced.

After testing and evaluating both navigation approaches, the user was asked to select his favorite method and explain why this method is better than the other one. In the end, the user was given the opportunity to write down additional feedback or improvement ideas for the two methods.

For a more detailed look at the questionnaire please refer to appendix A.3.3.

5.4 RESULTS

Thirty users took part in the user study, whereof 14 users already participated in the first study. To ensure equal conditions for both methods, the same number of experienced users started with the controller-based method and with the gesture-based one.

This section starts with an analysis of the users' demographic profile and continues with the subjective and objective assessments of the navigation methods. Thereby, the subjective assessment includes the users' difficulties with the methods, the users' condition and their final rating of the two methods. The objective assessment analyses the needed time to complete the last two levels and the number of drops from the runway in the third level. Afterward, the users' feedback and ideas are analyzed. In the end, all results and findings are summarized and presented in a concise overview.

As not all results can be covered in this section, additional charts and evaluations are included in appendix A.3.5.

5.4.1 DEMOGRAPHIC PROFILE

The 30 users were chosen from the students and staff of the TUM. As the pie chart in figure 5.7 shows, half of the participants were between 22 and 25 years old. Ten subjects were younger than 22 years and one was even younger than 18 years. From the five remaining users, two were older than 30 years.

The group of subjects included seven female and 23 male people. Therefrom, three people had already reached the master's degree, 11 people had reached the bachelor's degree or equivalent and 16 people had reached the high school degree. From the

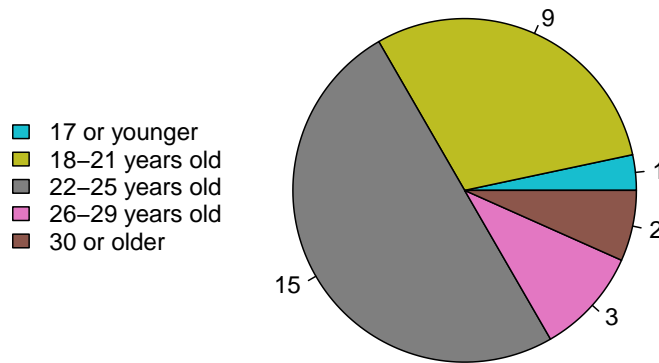


FIGURE 5.7: Pie chart showing the users' age.

subjects, only five peoples' main discipline is not computer science.

In terms of experience with VR headsets, 23.33 % stated, that they had never used a VR headset before and 60 % stated, that they use it less than one to three times a month. Regarding the Myo, 50 % of the participants said, that they already used the Myo before this user study and 40 % declared, that they did not know the Myo before. The other users stated, that although they did not use the Myo before, they did have heard of it.

Only eight people stated, that they did not use controllers, therefrom three people did not play games at all. From the remaining 22 users, four use a controller one to three times a month, five people use it more often and 12 people use a controller less often.

5.4.2 SUBJECTIVE ASSESSMENT OF THE NAVIGATION METHODS

The subjective assessment results from the evaluation of the questionnaire and is split into the users' difficulties with the methods, the precision of the methods, the users' condition and the ranking of the two methods.

The Users' Difficulties with the Methods: As shown in the diagram in figure 5.8 the users reported more problems with the gesture-based navigation method than with the controller-based one.

The greatest problem with the gesture-based method was the starting and stopping of a movements, which was reported by 12 users. This category also encompasses the toggling of the movement mode, which was, according to the users' feedback, the main source of this problem. Some users accidentally switched from the movement mode into the interaction mode just by putting down their arm, which was irritating for them. In addition to that, many users reported difficulties with the calibration of the Myo's forward direction when asked for the source of this problem.

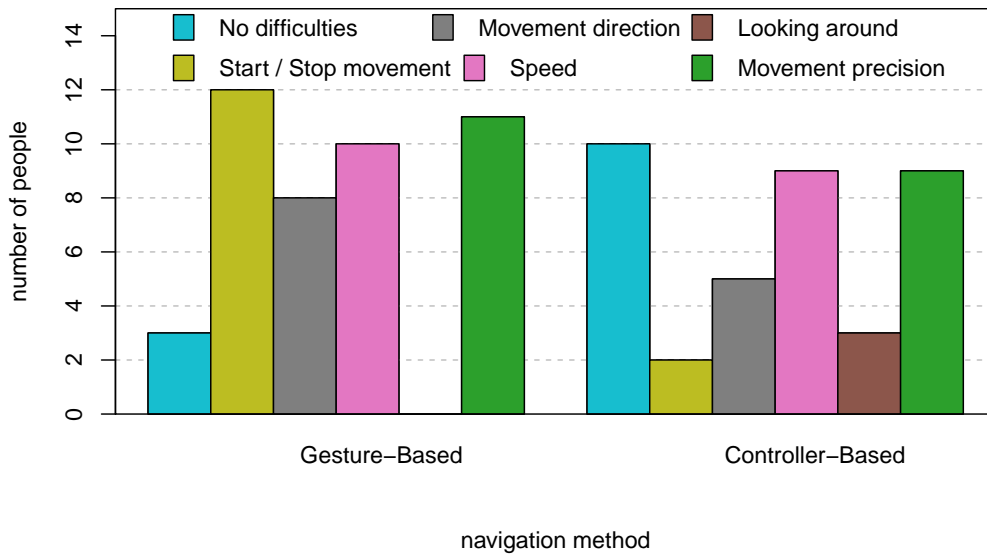


FIGURE 5.8: Bar chart depicting the users' assessment of the type of difficulty.

Beside these problems, 11 users reported difficulties with the movement precision of the gesture-based navigation method and nine with the controller-based method.

Another frequent problem reported for both navigation methods was the speed control which was too fast and too jerkily for one-third of the users. Thus, many users moved very fast through the levels, especially through the parkour, which caused some users to stagger a little bit.

In addition to that, about one-fourth of all users reported difficulties with defining the movement's direction with the gesture-based navigation method. In comparison, only 16 % of the subjects experienced the same problem with the controller-based method. In contrast to all other difficulties, the problem of looking around was only reported for the controller-based navigation method. This might be due to the fact, that most users who already used controllers before, were used to look around by utilizing the right joystick, which was not possible with the controller-based method used for this evaluation.

Precision of the Methods: As figure 5.9 shows, the assessment of the precision was not influenced by the experience of the users. The users which already took part in the first user study, rated the precision of the methods identically to the new users.

Both methods performed equally well with a rating of 3.6 and 3.7 respectively. Thus, the users stated, that the precision of the navigation methods was overall good.

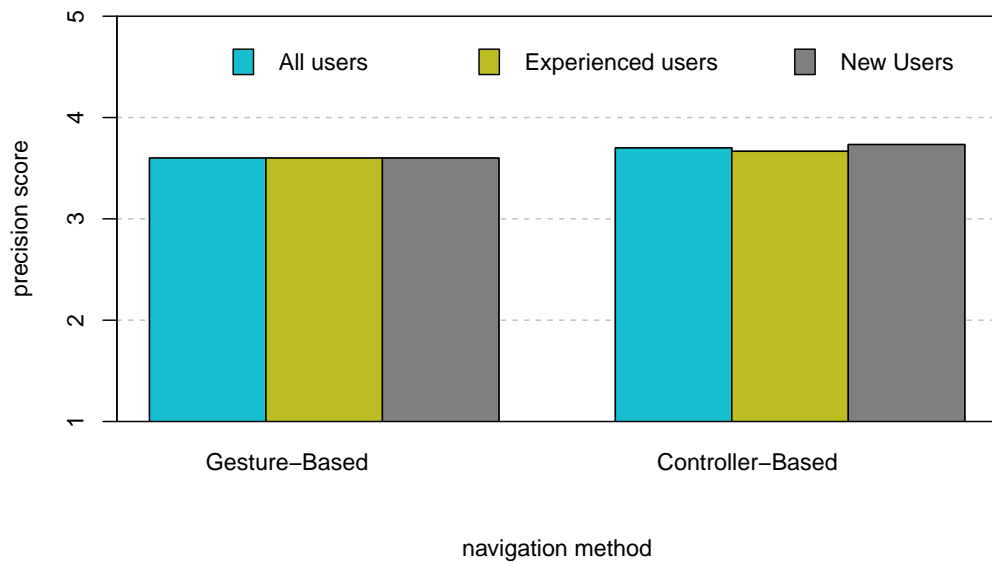


FIGURE 5.9: Bar chart depicting the users' assessment of the method's precision on a scale between 1 (very imprecise) and 5 (very precise).

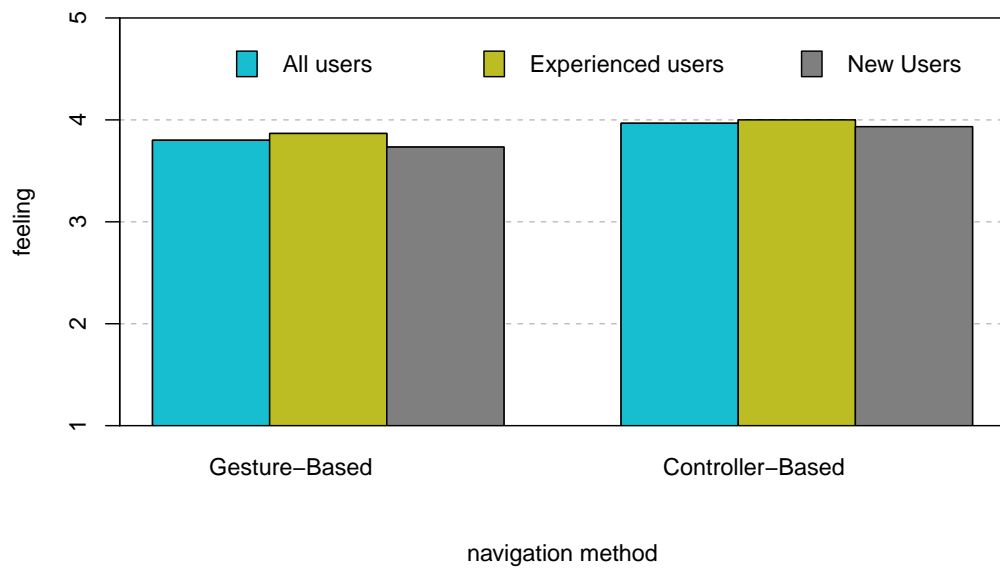


FIGURE 5.10: Bar chart depicting how the users felt while using the method on a scale from 1 (very bad) to 5 (very good).

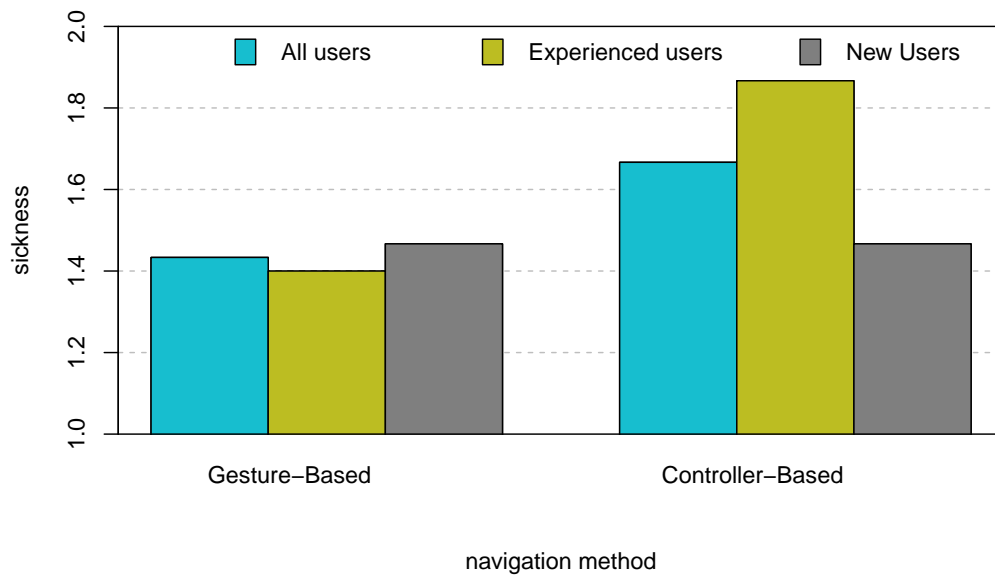


FIGURE 5.11: Bar chart visualizing the survey results regarding motion sickness on a scale from 1 (only a little dizzy) to 5 (very dizzy).

The Users' Condition: During the survey, the users were asked three questions regarding their condition while using the two navigation methods. These questions covered the feelings of the users while using the methods, their motion sickness, and their pain. The first question asked the users to specify how they felt while using the different navigation methods. To measure the users' feeling a scale from one to five was used, whereas higher values indicate a better feeling. As the chart in figure 5.10 shows, the users felt quite good while using the two methods. Moreover, no significant difference between the ratings of experienced and new users can be observed.

The next question asked the users to specify how dizzy or motion sick they felt during the study. Therefore, a scale from one to five was used, whereas, one indicated no dizziness and five a lot of dizziness. As figure 5.11 shows, the mean sickness score of the two methods is below 2, which is an improvement over the methods of the first user study. The mean rating of the gesture-based navigation method is with 1.43 better than the rating of the controller-based method with 1.67. For the controller-based method the mean rating of the experienced users is worse than the rating of the new users. Nevertheless, both methods can be considered reasonably good in terms of motion sickness as they both score low ratings.

The last question covered the pain caused by the two methods. From the 30 users, only 1 user reported a little bit pain while using the gesture-based navigation method. According to the user this pain was caused by the constant vibrations of the Myo, which indicated the movement. Compared to the last user study, this is a drastic improvement,

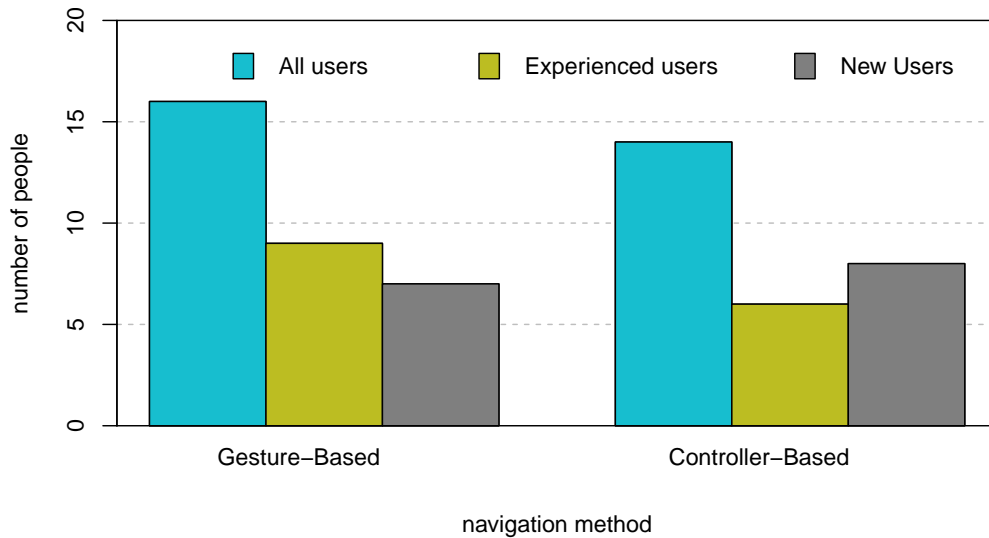


FIGURE 5.12: Visualization of the users' preferred navigation method.

which is most likely due to the abstinence of dedicated hand gestures and the reduced high the users need to raise their arm to.

The Rating of the Methods: In the last part of the questionnaire, the users were asked to select their preferred navigation method and explain why they prefer this method. The result of this question depicted in figure 5.12 shows that 53.3 % of all users prefer the gesture-based navigation method.

From the 16 people which chose the gesture-based method, a majority of 9 people (56.25 %) already took part in the first user study. From the experienced users a majority of 60 % favored the gesture-based method, whereas only 46.67 % of the new users shared this opinion. The other 53.33 % of new users selected the controller-based method, which forms the largest group of votes for this method.

When asked for their reasons for choosing the controller-based method, nearly all related users stated, that they were already accustomed to this type of navigation method and therefore, able to navigate with it better. Some users also had the feeling that the controller-based method was more precise when navigating along the small runway.

The users preferring the gesture-based method stated, that this method was more intuitive, they felt more integrated into the environment and the method itself felt more natural to move around. Some users also said that this method provided a more accurate speed and direction control. From the users who selected the gesture-based navigation method, many ensured that they would be able to improve their performance with this method quite quickly.

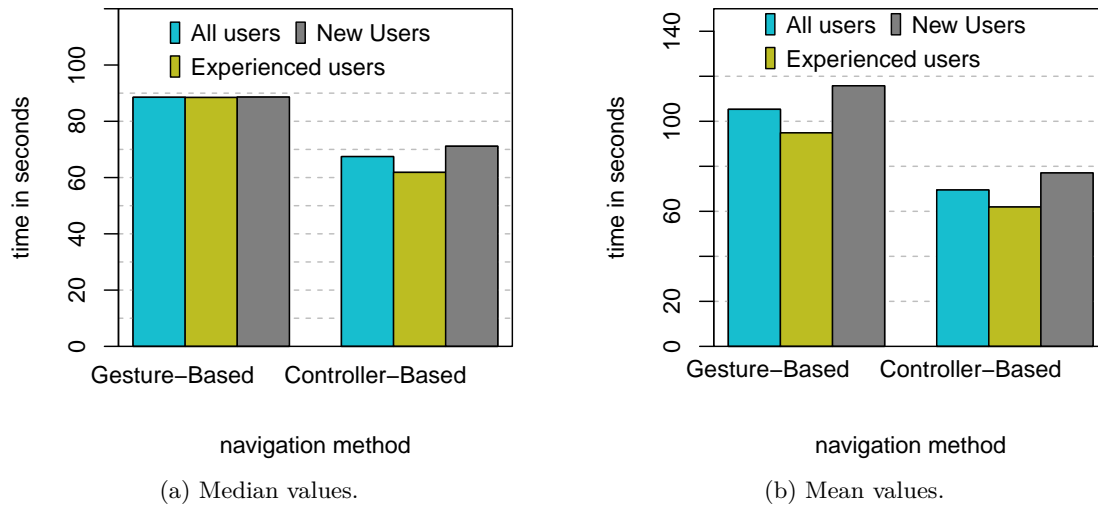


FIGURE 5.13: Visualization of two statistical measures representing the time needed to complete the parkour and the balance level.

5.4.3 OBJECTIVE ASSESSMENT OF THE NAVIGATION METHODS

The data acquired during the practical part of the survey resulted in the objective assessments mentioned in this section. The section is split into the acquired time data and the number of drops from the runway in the last level.

Time Data: Figure 5.13 shows the mean and median of the time needed to complete the parkour and the balance level together.

The mean time depicted in figure 5.13b shows, that the experienced users were able to finish the two levels faster with both methods. However, the median shows that the time differences between the two user groups can be neglected for the gesture-based navigation method as the experienced users were only 0.15 s ahead of the other users. A comparison of the median of all time values shows, that users are 23.81 % faster when using the controller-based navigation method. When looking at the mean time, users are even 34.02 % faster. This might be because most users are accustomed to pushing the joystick as far as possible to move forward in a computer game. Thus, the time values may suffer under the bias, that most people pushed the joystick all the way forward, while they did not do so with their arm.

This might also be the reason why many users reported that the maximal speed of the controller-based method was too fast, although it was equal to the maximal speed of the gesture-based method.

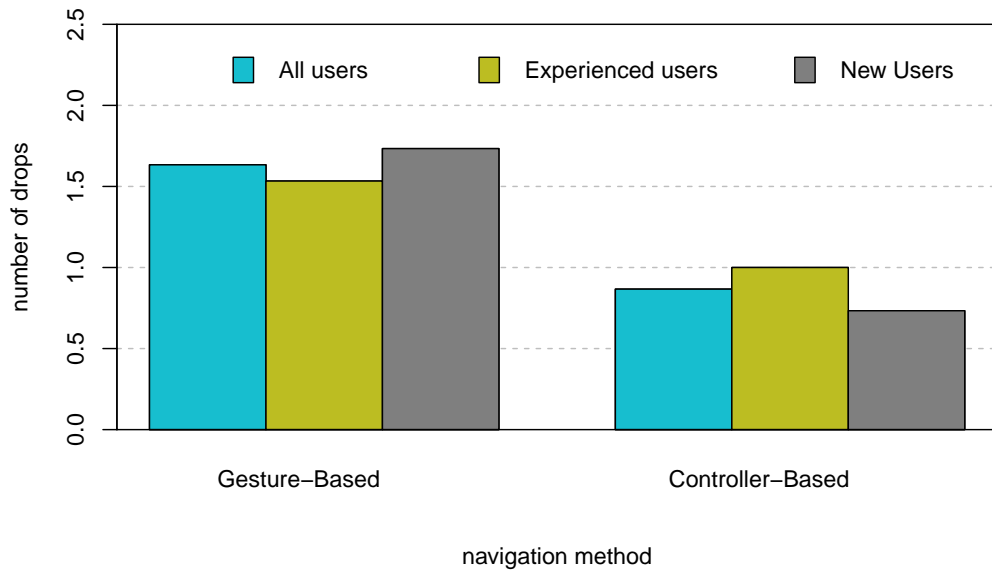


FIGURE 5.14: Illustration of the mean number of drops registered during the balance level.

Number of Drops: To measure the movement's precision the number of drops from the runway in the last level was recorded. Thereby, a lower number of drops is taken as an indication for more precise movements.

The chart depicted in figure 5.14 shows the mean number of drops, which is with 1.63 drops for the gesture-based method nearly twice as high as the number of drops for the controller-based method (0.87 drops). As the data contains some outliers, we took a look at the median, which is 1 drop for the gesture-based method and 0 drops for the controller-based one. Both statistical measures indicate, that the controller-based method was more precise than the gesture-based one.

Since many users stated, that the absence of feet in the simulation made balancing more difficult, it can be assumed that this aspect increased the number of drops. Taking this into consideration, both methods can be considered to be reasonably precise as the number of drops was quite small, when considering, that the runway had six turns.

5.4.4 AGGREGATED FEEDBACK

The users were given several opportunities to write down their feedback and ideas during the survey. Those comments were aggregated and are outlined here to give an overview of the users' thoughts and ideas.

The most frequent comment for both methods was that the maximal speed was too high.

Regarding the controller-based method many users stated, that the precise definition of the speed was difficult as the joystick's deflection range was too short for the large speed range. They also stated, that the speed acceleration was too fast.

Similar comments were made regarding the gesture-based method, where some users reported, that the speed mapping did not felt linear and was quite jerky. Some users stated that the threshold for starting a motion was too low so that they started moving unintended. However, other users said, that the threshold was not low enough as they did not start moving when slightly raising their arm. This shows, that an optimal mapping of speed to the arm's high is quite difficult and needs further research.

A problem which was also mentioned quite often for the gesture-based method was the toggling between the modes and the resulting calibration of the Myo. Many users stated, that they had a hard time, synchronizing the Myo correctly, as they were not able to see their arm in the virtual environment. They also mentioned, that they sometimes forgot to look straight while switching into the movement mode, which caused a wrong calibration of the Myo's forward direction. Some users reported, that they accidentally toggled the mode while putting the arm down to stop. In some cases this movement toggled the mode multiple times which lead to a sudden recalibration of the Myo. Thus, some users suggested, to use a specific gesture for toggling, which is not made by the users accidentally.

To help with the calibration, some users suggested to add a visual indication where the arm is pointing to and what mode is currently active.

The feedback regarding the vibrations during the movement was controversial. Some users stated, that it felt really uncomfortable and irritating as they were not able to distinguish between the vibrations signaling the toggling of modes and the vibrations indicating a movement. However, others said, that the vibrations were really useful and helped to distinguish between stopping and moving as well as during the toggling.

In contrast to the gesture-based navigation method, some users reported problems with looking around while using the controller-based method. Some users stated, that the combination of physical rotation and controller-based acceleration and direction definition was irritating as they were used to rotate with the right joystick when using a controller. However, others said that the combination was easy to learn and use. The first group of users suggested, to add the possibility to additionally turn with the right joystick, which may be especially useful when needing to make a 180° turn.

Many users stated, that the controller-based method felt more familiar and accurate.

However, it also made them feel less immersed in the virtual environment. Regarding the gesture-based method, many users said that it was intuitive to use and increased their feeling of immersion. They stated, that the method was quite easy to learn and they think, that they would be able to master it quickly.

As already mentioned, many users stated that precise navigation on the runway was difficult as they were not able to see their feet. One user suggested to solve this problem by adding a circle or cross right under the avatar to indicate where exactly the avatar is standing in the virtual space.

Another suggestion made by a user was to add the possibility to jump and step over objects to the set of available motions as he is always doing this in reality and thinks, that this would help to create a more natural navigation method.

In addition to the already mentioned remarks, the users also mentioned that the frame rate of the VR headset was quite low and that the simulation was sometimes jerky especially when moving fast.

5.4.5 SUMMARY AND IMPLICATIONS

Taking everything into consideration, it can be said that the two methods are equally well suited for navigation in VR.

Both methods scored similar ratings regarding the precision and the users' conditions. Although more users reported difficulties with the gesture-based navigation method, a small majority of 56.25 % declared that this method would be their favorite one. This might be due to the fact, that the controller-based method performed worse regarding the induced motion sickness.

However, during the objective assessment which consists of the time data and the mean number of drops, the controller-based method performed better. A possible explanation for this result might be the fact that 73.33 % of all users were already used to navigate with controllers.

The feedback gathered from the users did not only reveal which parts of the navigation method still needs improvement, but also that they see a lot of potential in the gesture-based navigation approach. They stated, that this method was easy to learn and use and felt intuitive, immersive and more natural for navigating in the virtual environment. The users also provided some suggestions for improvements like a smaller maximal speed, different gestures to toggle the modes and the addition of visual cues

to help the user with the correct calibration of the Myo's forward direction.

Overall it can be said, that the newly developed gesture-based navigation method performed equally well than the existing controller-based solutions, which shows, that this type of navigation provides a lot of potential. If further adjustments and improvements are included it might even be able to outperform traditional navigation methods.

CHAPTER 6

DISCUSSION AND CONCLUSION

This chapter provides a concise summary of this thesis and its findings as well as an overview of potential future work.

6.1 CONCLUSION

During this thesis a novel gesture-based navigation approach utilizing the Myo and the HTC Vive was developed. To find an appropriate method, we first designed different gesture-based navigation methods. From these methods, six were tested against each other in a user study, to evaluate their usability and discover their advantages and disadvantages.

Each method consisted of a different combination of `ControlType` and `NavigationType`, whereas the pitch of the user's arm was always used to control the speed. Thereby, the `ControlType` indicates which method is used to start and stop a movement and the `NavigationType` indicates which method is used to define the movement direction.

The utilized `ControlTypes` were called "StartStop" method, "HoldToMove" method and "ArmPitch" method. Whereas the first two methods used gestures to control the movements and the last method used the pitch of the user's arm.

The two `NavigationTypes` were the "ViveDirection" method and the "MyoDirection" method. For the first method, the viewing and moving direction are coupled together so that the user is only able to move into the direction he is looking at. The second method, decouples viewing and moving direction and allows the user to move independently from the viewing direction.

The first user study shows, that the gesture recognition capabilities of the Myo are not good enough to be used as the central part of a navigation method. As a result, the methods utilizing the "StartStop" and the "HoldToMove" control mechanism ranked badly in the subjective evaluation. Although, the "ArmPitch" method was favored by the users, the evaluation regarding pain showed, that all methods put a lot of stress on the users. Thereby, the users especially mentioned the pain caused by successively performing the same gesture and by holding up the arm for a longer period of time.

Regarding the `NavigationType`, the users did not agree on a favored method. Some users appreciated the decoupling between viewing and moving direction provided by the "MyoDirection" method. However, other users stated, that this decoupling made the navigation more difficult. Overall, most users ranked the "ArmPitch & MyoDirection" method first and the "ArmPitch & ViveDirection" method second.

The objective evaluation consisting of the time and the collision data confirmed the bad ratings of the "StartStop" and the "HoldToMove" methods. However, in contrast to the subjective ranking of the users, the objective data places the "ArmPitch & ViveDirection" method slightly better than the "ArmPitch & MyoDirection" method.

Although the subjective and objective evaluation agree on the "ArmPitch" as `ControlType`, they do not agree on the `NavigationType` as both types come with different advantages. The "ViveDirection" method is easier to learn and use and the "MyoDirection" method provides more possibilities as the viewing and moving direction are decoupled from each other.

With regard to the NRP, we decided to further develop the "ArmPitch & MyoDirection" method, as the users need to be able to perform a great variety of movements, for which the "MyoDirection" method is more suitable.

During the development of the method, some of the users' feedback regarding the speed mapping and the starting and stopping of movements was applied. Thus the initial position of the refined method is with the arm down, straight beneath the user's body. Raising the arm leads to a forward motion which reaches its maximal speed when the arm is outstretched in front of the user. Moving the arm backward, leads to a backward movement with fixed speed and direction.

As the proposed navigation method only includes basic movements, the method differs between an interaction and a movement mode to allow for an easy extension of the supported actions. To switch between these modes, any hand gesture can be used, as long as the arm is straight beneath the users' body.

Since the Myo needs to be calibrated every time the user switches into the movement mode, the user has to look straight forward during the toggling of the modes.

To analyze the overall potential of the refined method, a controller-based navigation approach was implemented as a reference for the evaluation. This method was designed according to established standards and uses the left joystick to control the user's movements and speed. The user's physical rotation controlled the avatar's rotation.

Both navigation methods were implemented in the Unity3D-Client and tested against each other in a second user study. Therefore, three simulation environments were designed on the NRP. The first environment served as a playground to get used to the two methods. The second environment was a parkour to test the methods' responsiveness. The third environment was a balance test which included a runway to test how precise the users can move with the methods.

The results of the final user study show that both methods are equally good in terms of precision and users' condition. According to the objective evaluation, the controller-based method is better in terms of completion time and number of drops from the runway. However, the gesture-based method has an advantage in terms of motion sickness and is favored by a majority of 53.3 % users. In addition to that, the users stated, that the gesture-based navigation method was easy to learn and intuitive, which is especially good with regards to the NRP as many non-computer scientists will have to use and learn this method.

The final user feedback revealed that many users perceived the maximum speed as too high. In addition to that many users stated that the toggling between the modes and the corresponding calibration of the Myo was quite challenging as the users were not able to see their arm and often forgot to look forward while toggling the mode. Thus it was suggested to add visual cues to make the calibration easier.

Although there is still much space for improvement, especially in terms of speed mapping, gesture recognition and calibration, our new navigation method has high potential and might even be able to surpass the controller-based approach in the future. Hence, our research question, which asked if the combination of Myo and HTC Vive would be able to create an immersive navigation method, can be definitely answered with yes.

6.2 FUTURE WORK

While this thesis has shown the development of an immersive gesture-based navigation method, there are still lots of other related issues that lie beyond the thesis's scope.

This section provides an overview of possible extensions to the proposed navigation method, the Unity3D-Client and additional research topics related to this work.

Mapping of Speed: Since a detailed evaluation of the speed control mechanism would have required a distinct user study, the research done in this area was not sufficient to find the perfect control mechanism. Thus many users reported difficulties with the mapping between speed and the pitch of the users' arm and also with the maximal speed level, which was too high for most users.

Therefore, a dedicated user study covering different speed control mechanisms, appropriate maximal speed levels and the deflection range of the user's arm needs to be performed. A resulting speed control mechanism should be precise and intuitive and needs to be included in the Unity3D-Client to improve the present method.

Vibration Feedback: As some users reported that the constant vibration on their arm felt uncomfortable and one user even stated that it felt painful, this feedback mechanism needs further improvements. However, the vibration cannot just be turned off, as many users stated that it was helpful, especially for the toggling of the modes. Thus another vibration frequency or feedback mechanism has to be found.

Calibration of the Myo: The calibration of the Myo's forward direction needs to be improved either by adding some visual cues or other feedback mechanisms. Thereby, it should be analyzed, how the calibration phase could be designed to allow the user to precisely calibrate the Myo without requiring too much time.

In addition to that, the calibration procedure should be changed so that the users do not have to calibrate the Myo every time the change into the movement mode. Therefore, one would need to take a look at the changes in the coordinate frame of the Camera induced by the head-tracking system of the HTC Vive as these changes cause the coordinate frames to diverge from each other if not in movement mode.

Toggling Between the Modes: The toggling between the modes needs to be improved, as some users accidentally triggered it just by putting down their arm.

This problem may be solved if the gesture recognition capabilities of the Myo are improved in a way, that a dedicated gesture can be used to toggle the mode. Therefore, the gesture classification of the Myo needs to be enhanced. One possibility to do so would be the application of different machine learning approaches.

Add Arms and Feet to the Avatar: During the two surveys many users mentioned that they would like to see their arms and feet in the virtual environment. Thus, the avatar in the NRP needs to be equipped with arms and preferably also with feet, to enhance the users spatial feeling. Therefore, not only the corresponding 3D-model but also the avatar's representation in ROS needs to be adapted.

If feet are not added to the model, a circle or something similar should be shown right under the avatar, to indicate, where exactly the user is standing. According to the users' feedback this would increase the movement's precision, especially when moving on small objects like the runway in the third level of the second user study.

In this scope, different types of cues and indications supporting the users' movements in virtual environments could be explored.

Network Delays: To prevent the last user study from suffering from potential network delays, a server next door was used to host the NRP. Since the HBP should foster the cooperation between researchers around the world it is very unlikely that the NRP will be used in such a setup. Thus, additional experiments with servers further away from the client should be made to evaluate the influence of network delays on the navigation in the NRP.

To diminish this influence, movement predictions could be included in the Unity3D-Client. However, when doing so, it should be taken into consideration, that the collision detection is solely done by the server so far.

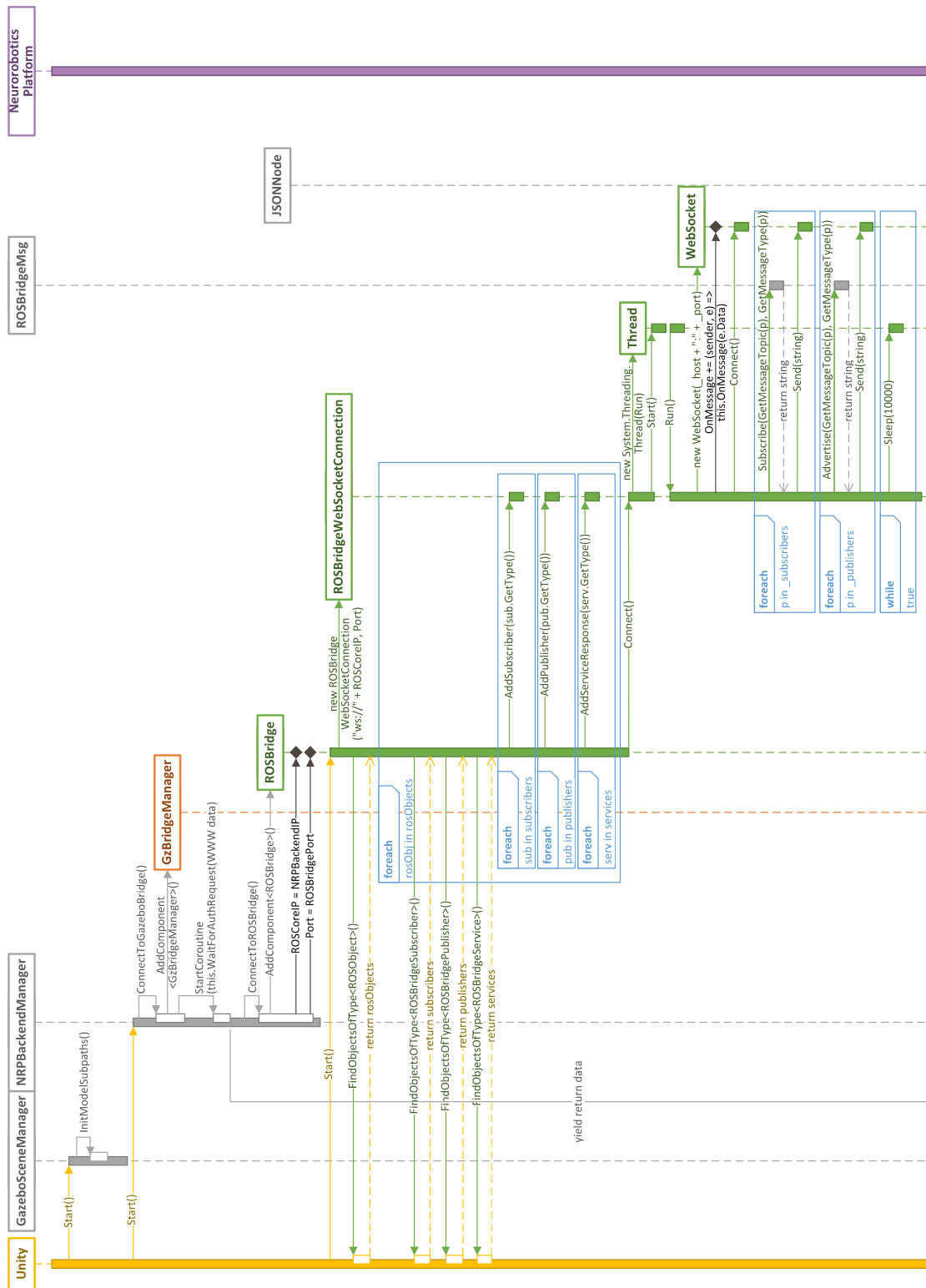
CHAPTER A

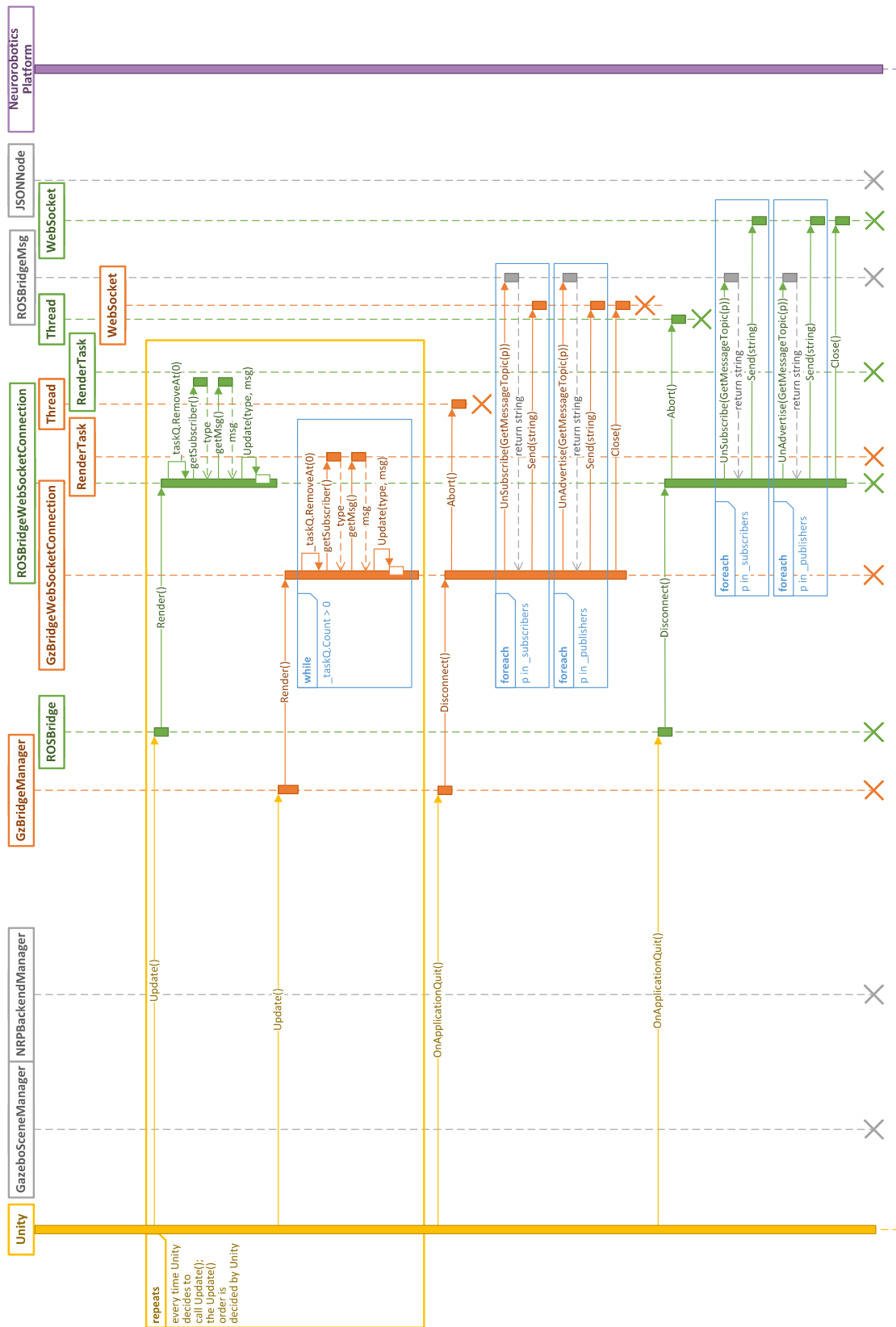
APPENDIX

A.1 UNITY3D-CLIENT SEQUENCE DIAGRAM

The following sequence diagram provides a detailed overview of the function calls and classes involved when communicating with the Neurorobotics Platform (NRP). The diagram depicts the code structure from the original Unity3D-Client developed by Sandro Weber and ignores the minor changes in the procedure done in the course of this thesis. Generally speaking the code can be logically divided into five different segments: System calls from Unity, incoming messages from the NRP, communication with Gazebo, communication with Robot Operating System (ROS) and general classes and functions. To facilitate the readability of this diagram, different colors were used for the different logical segments.

All calls from Unity, including the Start, Awake and Update calls are depicted in yellow. Calls from the NRP, which signal incoming messages from the different topics are depicted in violet. The gray classes and calls doesn't belong to a dedicated logical segment as they are needed by both, the communication establishment to the Gazebo bridge and the one to the ROS bridge. The communication with the Gazebo bridge is depicted in orange, whereas the communication with the ROS bridge is colored in green. Black arrows and the corresponding black diamonds symbolize the assignment of a public variable from another class.





A.2 USER STUDY: DIFFERENT NAVIGATION APPROACHES

This section includes all relevant materials concerning the first user study, where different Virtual Reality (VR) navigation approaches were compared against each other to find the most appropriate one.

A.2.1 FIGURES USED TO EXPLAIN THE DIFFERENT NAVIGATION APPROACHES

To explain the different navigation approaches to the user, the following figures were used. Thereby, the blue text on the upper half of the figure always provides information about the movement control mechanisms. On the right, upper side of the figure an avatar, symbolizing the user shows how the speed can be controlled and what kind of direction control mechanism is used. To emphasize the function of the direction control mechanism, the avatars on the bottom of the figure are used, whereas, the red arrow always points in the direction, the avatar would move when going forward.

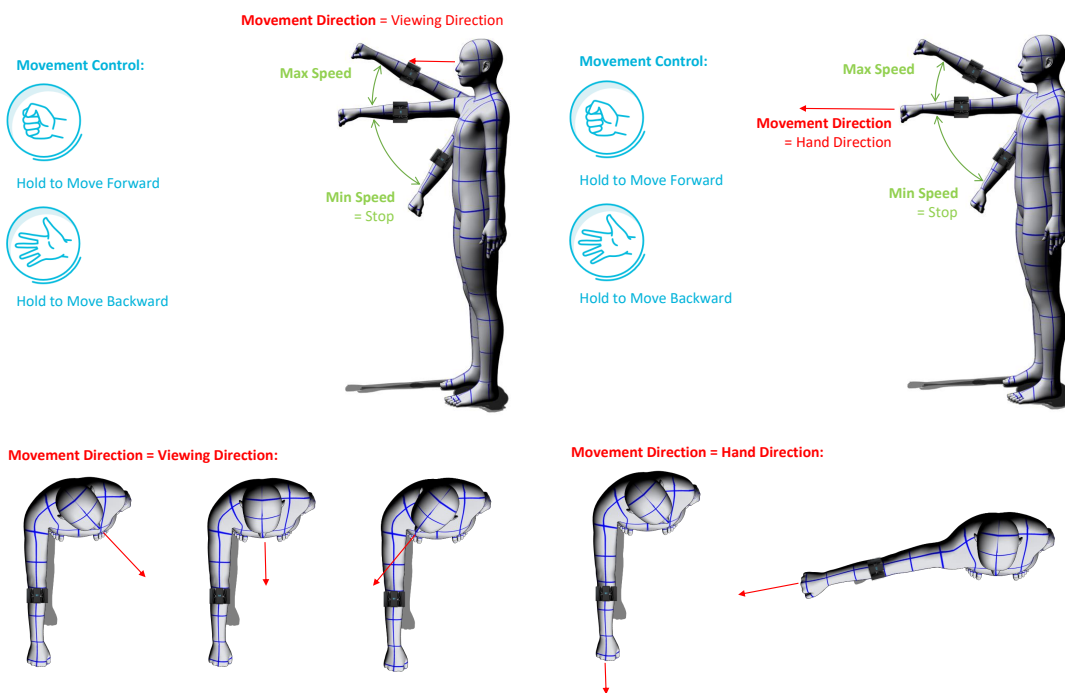


FIGURE A.1: Explanation for the method: Hold fist to move and gaze direction for steering

FIGURE A.2: Explanation for the method: Hold fist to move and hand direction for steering

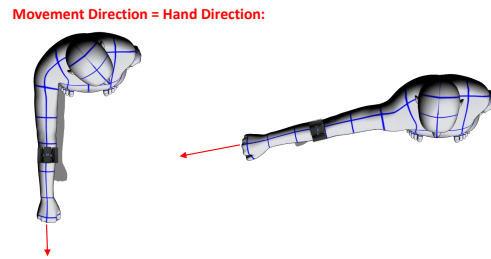
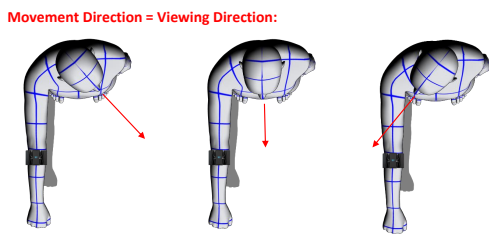
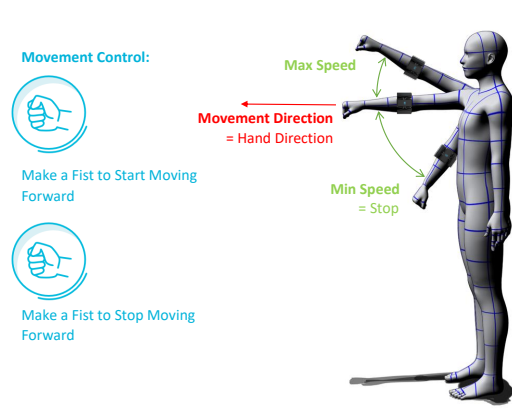
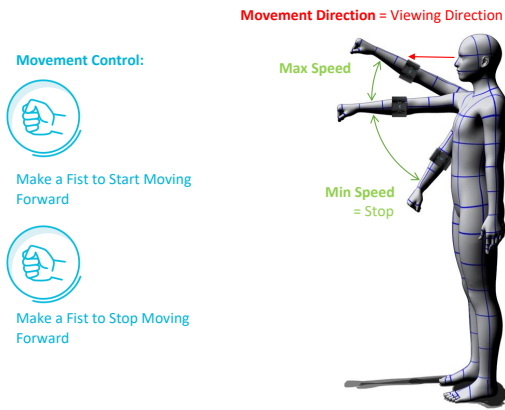


FIGURE A.3: Explanation for the method: Use start and stop gestures to move and gaze direction for steering

FIGURE A.4: Explanation for the method: Use start and stop gestures to move and gaze direction for steering

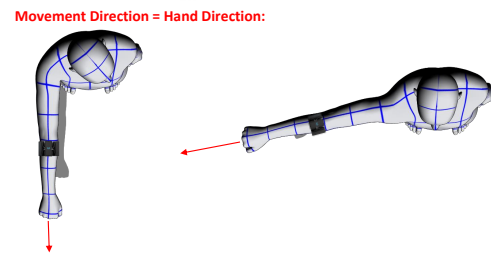
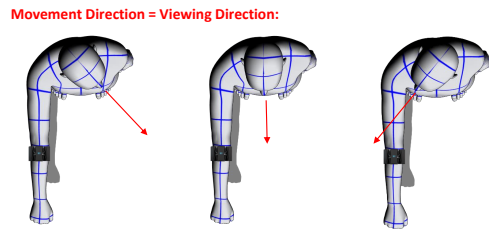
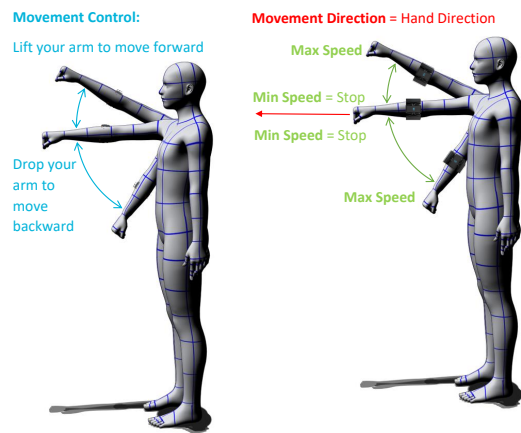
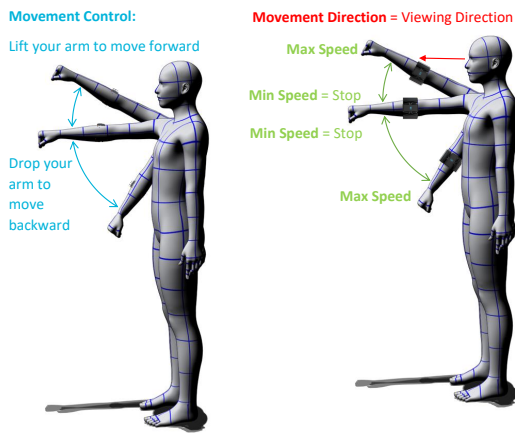


FIGURE A.5: Explanation for the method: Use the pitch of the arm to move and gaze direction for steering

FIGURE A.6: Explanation for the method: Use the pitch of the arm to move and gaze direction for steering

A.2.2 INFORMATION FOR THE USERS

The following information were given to all users who took place in the user study. As all users have German as their native language, the information was given in German to prevent misunderstandings due to language difficulties.

Summing up the following information, it briefly describes the background and incentive behind this thesis and the user study. In addition to that, the user is provided with information about the procedure of the study and what kind of data are raised in it's course. Lastly, the user is informed about potential risks and what to do in such cases.

Nutzerstudie zu unterschiedlichen Navigationsmethoden in VR

Informationen zur Studie für jeden Probanden

Zielsetzung der Arbeit

Der Titel der Masterarbeit lautet „Verbindung von Myo und HTC Vive für die Navigation in der Neurorobotics Platform“. Die Myo ist ein myoelektrisches Armband zur Gestensteuerung und die HTC Vive ist ein Virtual Reality Headset. Die Arbeit wird im Rahmen der Neurorobotics Platform erstellt. Diese ist ein Unterprojekt des Human Brain Projects, welches darauf abzielt, dass menschliche Gehirn besser zu verstehen. Die Neurorobotics Platform bietet dabei eine Simulationsumgebung für Roboter, welche mit virtuellen Gehirn-Modellen verknüpft werden. Da diese Roboter später auch in der Realität eingesetzt werden sollen, muss die Mensch-Roboter Interaktion gut funktionieren. Um diese simulieren und analysieren zu können, sollen die Menschen per Virtual Reality in die Simulation integriert werden. Damit die Interaktion so realistisch wie möglich ist, muss der Mensch in der Lage sein, sich möglichst einfach und intuitiv im virtuellen Raum bewegen zu können. Um das zu ermöglichen, sollen in dieser Studie unterschiedliche Navigationsmethoden getestet und basierend auf den Ergebnissen eine geeignete Methode ausgewählt werden.

Ablauf

- Die User Studie besteht sowohl aus einem praktischen Teil, in welchem die verschiedenen Methoden ausprobiert werden, als auch einem online Fragebogen, in welchem die Methoden bewertet werden sollen. Die Studie sollte zwischen 30 und 45 Minuten dauern und ist folgendermaßen aufgebaut:
- Als erstes muss sowohl die Myo als auch die HTC Vive eingestellt werden:

- Bei der Myo werden zuerst die Gesten auf der MyoDiagnostics getestet, damit der Proband weiß wie stark die Geste ausgeführt werden müssen. (http://diagnostics.myo.com/?utm_source=lesson-one&utm_medium=education&utm_campaign=myo-for-education)
- Bei der Vive muss der Riemen am Kopf richtig eingestellt werden.
- Danach wird der im praktischen Teil zu bewältigende Parkour anhand eines Videos vorgestellt. Dadurch soll der Lerneffekt ausgeglichen werden, welcher spätere Navigationsmethoden gegenüber den vorherigen Methoden bevorzugen würde.
- Im Anschluss wird mit dem ersten Teil des Fragebogens begonnen. Dieser besteht aus allgemeinen Fragen zu Alter, Geschlecht und Vorerfahrung. Diese Daten sind für die spätere Evaluierung nötig und werden über einen vom Probanden selbst generierten Code anonymisiert.
- Dann wird mit der ersten Navigationsmethode begonnen.
 - Diese wird zuerst anhand einer Zeichnung kurz vorgestellt.
 - Danach setzt der Proband die HTC Vive auf und startet zuerst auf einer offenen Fläche, wo er Gelegenheit hat die Navigationsmethode auszuprobieren.
 - Startposition ist dabei immer mit der rechten Hand gerade aus nach vorne gestreckt.
 - Wenn der Proband der Meinung ist, dass er die Methode ausreichend beherrscht, wechselt er über zweifaches tippen des Daumens und des Zeigefingers zu dem Parkour.
 - Dort muss der Proband dem Weg folgen und ins Ziel kommen. Der Proband wird gebeten, sich nicht die Umgebung anzusehen, sondern so schnell wie möglich ins Ziel zu gelangen.
 - Sobald das Ziel erreicht ist wird der Bildschirm blau und der Proband soll das Headset wieder ab-setzen.
- Im Anschluss kommt der Teil des Fragebogens welcher Informationen zu dieser Navigationsmethode sammelt.
- Danach folgt die nächste Navigationsmethode, welche wieder vorgestellt wird, danach folgt der praktische Teil und direkt danach kommt der Frageteil usw.

Insgesamt gibt es sechs Navigationsmethoden, welche alle vom Probanden getestet werden sollen.

- Nach den Fragen zur letzten Navigationsmethode folgen noch zwei weitere Fragen und dann ist die Studie beendet und der Proband wird über die erhobenen Daten informiert.

Datenverarbeitung und -erhebung

In der Studie werden folgende Daten erhoben:

- Bewegungsabläufe
- Zeiten
- Videomaterial von der HTC Vive

Alle Daten die während der Studie erhoben werden, werden anonymisiert und im Rahmen dieser Studie ausgewertet. Um die Korrektheit des Ergebnisses nachprüfbar zu machen, werden die anonymisierten Daten im Rahmen der Masterarbeit veröffentlicht. Erhobenes Videomaterial von der HTC Vive wird nur von mir, Tamara Barounig, persönlich ausgewertet und nicht an dritte weitergegeben oder gezeigt. Der Proband erklärt sich damit durch eine Unterschrift auf der Einverständniserklärung einverstanden.

Risiken

Im Laufe der Studie kann es im praktischen Teil zu Schwindelanfällen oder Übelkeit, sogenannter „Motion Sick-ness“ kommen. Der Proband möge hier daran erinnert sein, dass er die Studie nach eigenem Ermessen jederzeit abbrechen kann und sich auf keinen Fall Gefahren aussetzen sollte.

Zusätzliche Informationen

Zum Schluss soll noch einmal verdeutlicht werden, dass zu jedem Zeitpunkt der Studie lediglich die Methode und nicht der Proband selbst evaluiert wird. Jedwede Probleme oder Fehler die auftreten sind alleine durch die Methode und nicht durch den Probanden selbst verursacht.

A.2.3 QUESTIONNAIRE

The survey was conducted with a self-hosted LimeSurvey instance and the following questionnaire is the printable version of the survey the users were asked to complete after each navigation method.

Thank you for taking part in this Study!

Section A: Preliminary Questions

A1.
Please fill in your self generated identifier. The identifier is a 7 digit code which looks as follows: NFMNBA
NF: The first and second letter of your father's first name in capital letters. MM: The month (01 - 12) in which your mother was born. NB: The first and second letter of your birthplace in capital letters. A: Does your own LAST name begin with a letter in the first (1) half of the alphabet (A - M) or in the second (2) half of the alphabet (N - Z). Please write 1 or 2 respectively.

A2. What is your age?

17 or younger
18-21 years old
22-25 years old
26-29 years old
30 or older

A3. To which gender identity do you most identify?

Female
Male
Other

Other

A9. How often did you use the Myo?

One to three times a month
One to three times a week
More than three times a week

Section B: Practical Part 1/6

B1. Did you experience any difficulties or problems with the navigation method?

No difficulties at all
Some difficulties, which I don't deem crucial
Some difficulties, which I think effect the navigation experience
Some problems, which make the navigation method unusable

B2. To which of the following categories would you assign your difficulties or problems?

Starting and stopping the movement
Defining the movement direction
Defining the speed
Looking around

B3. Could you please explain the difficulties or problems in more detail?

B4. How did the proposed method made you feel?
(1 = very bad; 5 = very good)

1
2
3
4
5

A4. What is the highest degree or level of school you have completed?
If you're currently enrolled in school, please indicate the highest degree you have received.

High school degree or equivalent
Bachelor's degree (BA)
Diploma's degree (Dipl.)
Master's degree (MA)
Doctorate (PhD)

A5. What is your main discipline?
If you are a student, please select your current subject. If you are employed, please select your working area.

Informatics
Mathematics
Other

Other

A6. How often did you use a Virtual Reality Headset in the last six month?

Never
One to three times a month
One to three times a week
More than three times a week

A7. Was the headset you used the HTC Vive?

Yes
No

A8. Did you know the Myo armband before this experiment?

No
Yes, I heard from it
Yes, I used it myself

B5. Did the method let you experience any motion sickness or dizziness?

Yes
No

B6. How dizzy or motion sick let the method feel you?
(1 = only a little dizzy; 5 = very dizzy)

1
2
3
4
5

B7. Did the method caused you any pain in the hands?

Yes
No

B8. How much pain did you feel?
(1 = only a little; 5 = very much)

1
2
3
4
5

B9. Please specify where you feel pain.

B10. Any comments to the presented navigation technique?

A.2 USER STUDY: DIFFERENT NAVIGATION APPROACHES

Section C: Practical Part 2/6

C1. Did you experience any difficulties or problems with the navigation method?

No difficulties at all

Some difficulties, which I don't deem crucial

Some difficulties, which I think effect the navigation experience

Some problems, which make the navigation method unusable

C2. To which of the following categories would you assign your difficulties or problems?

Starting and stopping the movement

Defining the movement direction

Defining the speed

Looking around

C3. Could you please explain the difficulties or problems in more detail?

C4. How did the proposed method made you feel?

(1 = very bad, 5 = very good)

1

2

3

4

5

C5. Did the method let you experience any motion sickness or dizziness?

Yes

No

C6. How dizzy or motion sick let the method feel you?

(1 = only a little dizzy, 5 = very dizzy)

1

2

3

4

5

C7. Did the method caused you any pain in the hands?

Yes

No

C8. How much pain did you feel?

(1 = only a little, 5 = very much)

1

2

3

4

5

C9. Please specify where you feel pain.

C10. Any comments to the presented navigation technique?

Section D: Practical Part 3/6

D1. Did you experience any difficulties or problems with the navigation method?

No difficulties at all

Some difficulties, which I don't deem crucial

Some difficulties, which I think effect the navigation experience

Some problems, which make the navigation method unusable

D2. To which of the following categories would you assign your difficulties or problems?

Starting and stopping the movement

Defining the movement direction

Defining the speed

Looking around

D3. Could you please explain the difficulties or problems in more detail?

D4. How did the proposed method made you feel?

(1 = very bad, 5 = very good)

1

2

3

4

5

D5. Did the method let you experience any motion sickness or dizziness?

Yes

No

D6. How dizzy or motion sick let the method feel you?

(1 = only a little dizzy, 5 = very dizzy)

1

2

3

4

5

D7. Did the method caused you any pain in the hands?

Yes

No

D8. How much pain did you feel?

(1 = only a little, 5 = very much)

1

2

3

4

5

D9. Please specify where you feel pain.

D10. Any comments to the presented navigation technique?

CHAPTER A: APPENDIX

Section E: Practical Part 4/6

E1. Did you experience any difficulties or problems with the navigation method?

No difficulties at all

Some difficulties, which I don't deem crucial

Some difficulties, which I think effect the navigation experience

Some problems, which make the navigation method unusable

E2. To which of the following categories would you assign your difficulties or problems?

Starting and stopping the movement

Defining the movement direction

Defining the speed

Looking around

E3. Could you please explain the difficulties or problems in more detail?

E4. How did the proposed method made you feel?

(1 = very bad; 5 = very good)

1

2

3

4

5

E5. Did the method let you experience any motion sickness or dizziness?

Yes

No

E6. How dizzy or motion sick let the method feel you?

(1 = only a little dizzy; 5 = very dizzy)

1

2

3

4

5

E7. Did the method caused you any pain in the hands?

Yes

No

E8. How much pain did you feel?

(1 = only a little; 5 = very much)

1

2

3

4

5

E9. Please specify where you feel pain.

E10. Any comments to the presented navigation technique?

Section F: Practical Part 5/6

F1. Did you experience any difficulties or problems with the navigation method?

No difficulties at all

Some difficulties, which I don't deem crucial

Some difficulties, which I think effect the navigation experience

Some problems, which make the navigation method unusable

F2. To which of the following categories would you assign your difficulties or problems?

Starting and stopping the movement

Defining the movement direction

Defining the speed

Looking around

F3. Could you please explain the difficulties or problems in more detail?

F4. How did the proposed method made you feel?

(1 = very bad; 5 = very good)

1

2

3

4

5

F5. Did the method let you experience any motion sickness or dizziness?

Yes

No

F6. How dizzy or motion sick let the method feel you?

(1 = only a little dizzy; 5 = very dizzy)

1

2

3

4

5

F7. Did the method caused you any pain in the hands?

Yes

No

F8. How much pain did you feel?

(1 = only a little; 5 = very much)

1

2

3

4

5

F9. Please specify where you feel pain.

F10. Any comments to the presented navigation technique?

A.2 USER STUDY: DIFFERENT NAVIGATION APPROACHES

Section G: Practical Part 6/6

G1. Did you experience any difficulties or problems with the navigation method?

No difficulties at all

Some difficulties, which I don't deem crucial

Some difficulties, which I think effect the navigation experience

Some problems, which make the navigation method unusable

G2. To which of the following categories would you assign your difficulties or problems?

Starting and stopping the movement

Defining the movement direction

Defining the speed

Looking around

G3. Could you please explain the difficulties or problems in more detail?

G4. How did the proposed method made you feel?

(1 = very bad, 5 = very good)

1

2

3

4

5

G5. Did the method let you experience any motion sickness or dizziness?

Yes

No

G6. How dizzy or motion sick let the method feel you?

(1 = only a little dizzy, 5 = very dizzy)

1

2

3

4

5

G7. Did the method caused you any pain in the hands?

Yes

No

G8. How much pain did you feel?

(1 = only a little, 5 = very much)

1

2

3

4

5

G9. Please specify where you feel pain.

G10. Any comments to the presented navigation technique?

Section H: Final Questions

H1. Please rank the methods in the order you think they are most suitable for navigation in VR.

Hold fist to move & gaze direction for steering

Hold fist to move & hand direction for steering

Use start and stop gestures to move & gaze direction for steering

Use start and stop gestures to move & hand direction for steering

Use the pitch of the arm to move & hand direction for steering

Use the pitch of the arm to move & gaze direction for steering

H2. Please provide any critical or positive feedback if you like, as both is very valuable for the study.

Thank you very much! You saved my Master thesis!

A.2.4 DATA GATHERED DURING THE PRACTICAL PART

During the practical part of the user study, the time needed to get used to the navigation method, the time needed to complete the parkour and the number of collisions registered during the parkour were saved in a Comma-Separated Values (CSV) file. As the order of the tested navigation methods differs from user to user, the order and the corresponding method are also saved for each user. The resulting raw data can be seen in the following table.

Identifier	Order	ContrType	NavType	Time [s] TestArea	Time [s] Parkour	Number Collisions
AL02FI1	1	StartStop	ViveDirection	71.262	48.911	233
AL02FI1	2	ArmPitch	ViveDirection	49.658	48.878	13
AL02FI1	3	ArmPitch	MyoDirection	55.790	48.611	66
AL02FI1	4	HoldToMove	MyoDirection	56.871	41.656	36
AL02FI1	5	HoldToMove	ViveDirection	32.369	39.300	4
AL02FI1	6	StartStop	MyoDirection	55.354	36.711	65
AL09AU1	1	ArmPitch	MyoDirection	56.088	61.211	6
AL09AU1	2	StartStop	ViveDirection	32.466	44.978	23
AL09AU1	3	StartStop	MyoDirection	33.282	52.200	31
AL09AU1	4	HoldToMove	ViveDirection	52.933	42.278	15
AL09AU1	5	HoldToMove	MyoDirection	22.404	54.567	54
AL09AU1	6	ArmPitch	ViveDirection	27.890	36.778	1
AL12FR2	1	StartStop	ViveDirection	50.165	59.656	53
AL12FR2	2	ArmPitch	MyoDirection	68.822	41.078	21
AL12FR2	3	HoldToMove	ViveDirection	87.384	41.344	11
AL12FR2	4	ArmPitch	ViveDirection	63.265	37.578	13
AL12FR2	5	HoldToMove	MyoDirection	64.487	37.067	55
AL12FR2	6	StartStop	MyoDirection	42.917	27.678	8
AN10RO1	1	StartStop	ViveDirection	21.234	47.733	166
AN10RO1	2	ArmPitch	ViveDirection	64.458	47.244	24
AN10RO1	3	HoldToMove	ViveDirection	24.511	58.745	25
AN10RO1	4	HoldToMove	MyoDirection	11.374	75.645	0
AN10RO1	5	ArmPitch	MyoDirection	52.770	80.711	18
AN10RO1	6	StartStop	MyoDirection	40.416	84.989	267
CH01MU2	1	HoldToMove	ViveDirection	91.065	71.489	1123
CH01MU2	2	HoldToMove	MyoDirection	48.600	39.644	1
CH01MU2	3	StartStop	ViveDirection	37.069	71.956	55
CH01MU2	4	ArmPitch	ViveDirection	37.985	41.533	5
CH01MU2	5	ArmPitch	MyoDirection	30.504	41.967	5
CH01MU2	6	StartStop	MyoDirection	31.704	50.989	1
FE01FR1	1	HoldToMove	MyoDirection	66.895	120.944	352
FE01FR1	2	HoldToMove	ViveDirection	25.301	61.956	167
FE01FR1	3	StartStop	ViveDirection	30.663	70.133	213
FE01FR1	4	StartStop	MyoDirection	22.563	56.456	0
FE01FR1	5	ArmPitch	ViveDirection	36.362	47.044	43
FE01FR1	6	ArmPitch	MyoDirection	39.452	46.800	10
FI12RO2	1	HoldToMove	MyoDirection	38.850	47.256	11

A.2 USER STUDY: DIFFERENT NAVIGATION APPROACHES

Identifier	Order	ContrType	NavType	Time [s]	Time [s]	Number Collisions
				TestArea	Parkour	
FI12RO2	2	ArmPitch	ViveDirection	57.266	81.922	48
FI12RO2	3	ArmPitch	MyoDirection	50.933	47.300	461
FI12RO2	4	StartStop	ViveDirection	22.905	112.922	142
FI12RO2	5	StartStop	MyoDirection	20.404	57.400	184
FI12RO2	6	HoldToMove	ViveDirection	40.535	74.000	187
FR11MU2	1	ArmPitch	MyoDirection	34.042	53.311	136
FR11MU2	2	HoldToMove	MyoDirection	43.754	47.011	197
FR11MU2	3	StartStop	MyoDirection	43.108	43.200	142
FR11MU2	4	HoldToMove	ViveDirection	59.976	65.800	150
FR11MU2	5	StartStop	ViveDirection	47.858	38.267	57
FR11MU2	6	ArmPitch	ViveDirection	71.323	39.611	31
GE01MU1	1	StartStop	ViveDirection	48.414	42.844	40
GE01MU1	2	StartStop	MyoDirection	23.763	32.600	23
GE01MU1	3	ArmPitch	MyoDirection	24.279	35.733	5
GE01MU1	4	ArmPitch	ViveDirection	46.362	30.011	28
GE01MU1	5	HoldToMove	MyoDirection	21.541	31.100	15
GE01MU1	6	HoldToMove	ViveDirection	21.119	25.400	0
GE01RE1	1	ArmPitch	ViveDirection	84.516	39.278	172
GE01RE1	2	HoldToMove	MyoDirection	57.517	33.689	190
GE01RE1	3	HoldToMove	ViveDirection	56.128	27.733	69
GE01RE1	4	StartStop	MyoDirection	58.244	34.700	112
GE01RE1	5	ArmPitch	MyoDirection	79.120	36.133	94
GE01RE1	6	StartStop	ViveDirection	55.686	29.745	101
HA02MU1	1	StartStop	MyoDirection	111.691	54.511	380
HA02MU1	2	HoldToMove	MyoDirection	43.999	35.056	78
HA02MU1	3	ArmPitch	ViveDirection	21.647	33.456	64
HA02MU1	4	ArmPitch	MyoDirection	61.149	32.589	9
HA02MU1	5	HoldToMove	ViveDirection	41.494	30.800	66
HA02MU1	6	StartStop	ViveDirection	25.496	41.022	161
HE02MU2	1	ArmPitch	ViveDirection	86.783	76.911	25
HE02MU2	2	StartStop	MyoDirection	39.224	53.622	7
HE02MU2	3	HoldToMove	MyoDirection	38.985	63.856	155
HE02MU2	4	ArmPitch	MyoDirection	41.607	66.267	73
HE02MU2	5	StartStop	ViveDirection	43.230	42.889	24
HE02MU2	6	HoldToMove	ViveDirection	47.206	92.067	158
HO06AU1	1	ArmPitch	MyoDirection	96.462	97.322	12
HO06AU1	2	HoldToMove	MyoDirection	68.324	61.767	3
HO06AU1	3	StartStop	ViveDirection	21.693	94.367	7
HO06AU1	4	StartStop	MyoDirection	37.837	48.589	2
HO06AU1	5	HoldToMove	ViveDirection	47.140	58.456	58
HO06AU1	6	ArmPitch	ViveDirection	39.835	51.633	18
JO06GR1	1	ArmPitch	ViveDirection	80.853	57.544	86
JO06GR1	2	StartStop	ViveDirection	24.146	333.145	192
JO06GR1	3	HoldToMove	MyoDirection	52.291	183.067	98
JO06GR1	4	StartStop	MyoDirection	25.418	118.689	234
JO06GR1	5	ArmPitch	MyoDirection	53.240	54.111	10
JO06GR1	6	HoldToMove	ViveDirection	42.716	49.044	81

CHAPTER A: APPENDIX

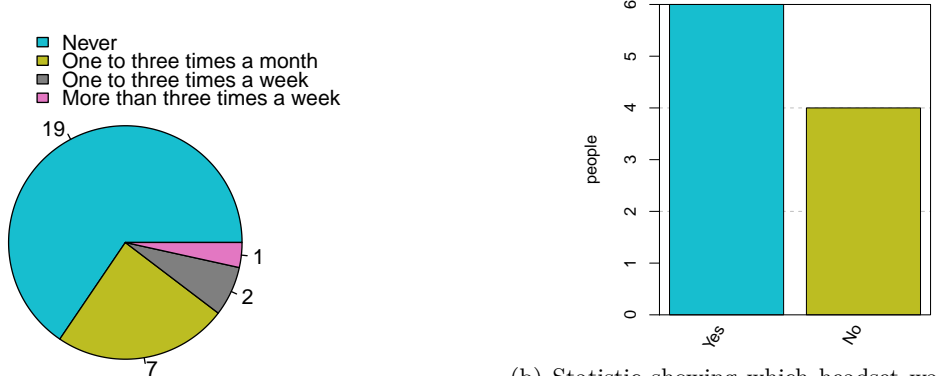
Identifier	Order	ContrType	NavType	Time [s]	Time [s]	Number Collisions
				TestArea	Parkour	
JO07MA1	1	StartStop	MyoDirection	42.302	80.556	231
JO07MA1	2	HoldToMove	ViveDirection	24.553	51.744	44
JO07MA1	3	HoldToMove	MyoDirection	20.458	58.467	54
JO07MA1	4	StartStop	ViveDirection	35.643	70.333	80
JO07MA1	5	ArmPitch	ViveDirection	28.085	34.478	7
JO07MA1	6	ArmPitch	MyoDirection	26.290	41.967	4
JO08CO2	1	HoldToMove	ViveDirection	40.665	33.511	98
JO08CO2	2	StartStop	MyoDirection	28.572	39.678	2
JO08CO2	3	ArmPitch	ViveDirection	22.410	30.100	60
JO08CO2	4	ArmPitch	MyoDirection	31.718	56.867	121
JO08CO2	5	StartStop	ViveDirection	57.153	41.144	50
JO08CO2	6	HoldToMove	MyoDirection	36.550	30.400	47
JU08ME1	1	HoldToMove	MyoDirection	47.333	154.178	738
JU08ME1	2	ArmPitch	MyoDirection	29.875	41.411	117
JU08ME1	3	StartStop	MyoDirection	88.463	39.800	76
JU08ME1	4	ArmPitch	ViveDirection	25.968	32.200	1
JU08ME1	5	HoldToMove	ViveDirection	42.363	73.678	269
JU08ME1	6	StartStop	ViveDirection	20.484	29.856	28
MA10MU2	1	StartStop	MyoDirection	61.671	46.267	0
MA10MU2	2	HoldToMove	ViveDirection	26.381	36.811	27
MA10MU2	3	HoldToMove	MyoDirection	32.039	36.422	29
MA10MU2	4	ArmPitch	MyoDirection	49.150	33.289	28
MA10MU2	5	StartStop	ViveDirection	48.207	29.100	2
MA10MU2	6	ArmPitch	ViveDirection	62.832	35.289	24
MI07WO1	1	ArmPitch	MyoDirection	70.108	48.511	44
MI07WO1	2	StartStop	MyoDirection	50.087	79.889	723
MI07WO1	3	ArmPitch	ViveDirection	52.869	47.556	30
MI07WO1	4	StartStop	ViveDirection	38.758	46.722	30
MI07WO1	5	HoldToMove	ViveDirection	57.040	38.000	7
MI07WO1	6	HoldToMove	MyoDirection	69.727	57.300	50
OT12WA1	1	StartStop	MyoDirection	66.718	68.289	120
OT12WA1	2	StartStop	ViveDirection	73.753	49.711	30
OT12WA1	3	ArmPitch	MyoDirection	89.396	50.200	51
OT12WA1	4	HoldToMove	ViveDirection	37.424	46.867	19
OT12WA1	5	ArmPitch	ViveDirection	71.928	34.967	73
OT12WA1	6	HoldToMove	MyoDirection	47.833	54.800	96
PA03CE2	1	ArmPitch	MyoDirection	77.166	55.500	159
PA03CE2	2	StartStop	ViveDirection	28.740	52.178	140
PA03CE2	3	HoldToMove	ViveDirection	21.703	39.289	25
PA03CE2	4	StartStop	MyoDirection	35.044	33.400	36
PA03CE2	5	HoldToMove	MyoDirection	35.808	40.178	102
PA03CE2	6	ArmPitch	ViveDirection	51.385	27.622	44
PE11MU1	1	HoldToMove	ViveDirection	36.998	71.289	193
PE11MU1	2	ArmPitch	ViveDirection	47.573	43.544	38
PE11MU1	3	StartStop	MyoDirection	87.979	39.322	151
PE11MU1	4	StartStop	ViveDirection	31.471	37.444	37
PE11MU1	5	HoldToMove	MyoDirection	59.436	36.889	179

A.2 USER STUDY: DIFFERENT NAVIGATION APPROACHES

Identifier	Order	ContrType	NavType	Time [s]	Time [s]	Number Collisions
				TestArea	Parkour	
PE11MU1	6	ArmPitch	MyoDirection	74.585	32.456	58
RA07FR1	1	ArmPitch	ViveDirection	62.557	45.311	268
RA07FR1	2	HoldToMove	ViveDirection	38.280	38.900	94
RA07FR1	3	HoldToMove	MyoDirection	29.484	30.278	218
RA07FR1	4	ArmPitch	MyoDirection	19.862	33.611	28
RA07FR1	5	StartStop	ViveDirection	47.724	37.533	165
RA07FR1	6	StartStop	MyoDirection	23.327	30.911	0
RE11HA1	1	StartStop	MyoDirection	23.117	68.056	244
RE11HA1	2	ArmPitch	MyoDirection	38.876	42.011	83
RE11HA1	3	HoldToMove	ViveDirection	50.328	51.178	125
RE11HA1	4	StartStop	ViveDirection	30.883	46.933	37
RE11HA1	5	ArmPitch	ViveDirection	43.486	36.200	25
RE11HA1	6	HoldToMove	MyoDirection	58.350	39.078	45
RO06LI2	1	StartStop	ViveDirection	40.660	77.000	92
RO06LI2	2	ArmPitch	MyoDirection	30.254	65.689	25
RO06LI2	3	HoldToMove	ViveDirection	29.389	59.022	0
RO06LI2	4	ArmPitch	ViveDirection	23.487	54.200	27
RO06LI2	5	StartStop	MyoDirection	35.598	54.711	28
RO06LI2	6	HoldToMove	MyoDirection	14.427	51.933	44
RO09DA1	1	HoldToMove	ViveDirection	49.562	47.878	61
RO09DA1	2	ArmPitch	ViveDirection	32.850	32.322	119
RO09DA1	3	StartStop	ViveDirection	34.995	32.778	60
RO09DA1	4	HoldToMove	MyoDirection	44.940	39.367	28
RO09DA1	5	StartStop	MyoDirection	20.062	36.455	49
RO09DA1	6	ArmPitch	MyoDirection	29.355	30.478	20
VO12BA2	1	ArmPitch	ViveDirection	25.233	53.411	34
VO12BA2	2	StartStop	ViveDirection	29.935	57.356	88
VO12BA2	3	ArmPitch	MyoDirection	53.187	41.667	85
VO12BA2	4	HoldToMove	MyoDirection	66.015	34.645	223
VO12BA2	5	StartStop	MyoDirection	48.214	52.733	170
VO12BA2	6	HoldToMove	ViveDirection	44.562	41.556	65
WE06AU1	1	HoldToMove	MyoDirection	71.572	84.744	591
WE06AU1	2	ArmPitch	MyoDirection	56.039	59.456	29
WE06AU1	3	StartStop	MyoDirection	68.045	60.678	33
WE06AU1	4	HoldToMove	ViveDirection	25.752	51.156	43
WE06AU1	5	ArmPitch	ViveDirection	63.417	53.600	17
WE06AU1	6	StartStop	ViveDirection	40.824	51.622	0
WO04AU2	1	HoldToMove	ViveDirection	109.129	63.733	37
WO04AU2	2	StartStop	MyoDirection	60.919	59.311	153
WO04AU2	3	ArmPitch	ViveDirection	55.149	37.033	79
WO04AU2	4	HoldToMove	MyoDirection	47.606	43.989	52
WO04AU2	5	ArmPitch	MyoDirection	57.828	34.222	27
WO04AU2	6	StartStop	ViveDirection	28.115	33.678	65

A.2.5 DETAILED SURVEY RESULTS

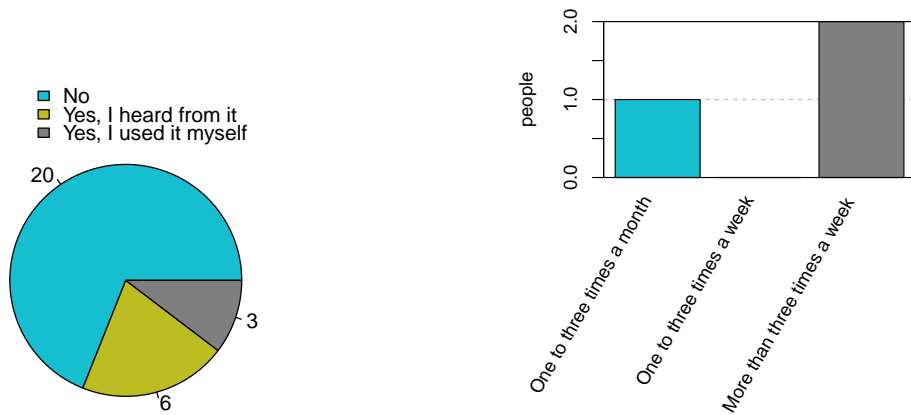
This section includes additional survey results not addressed in the main part of this thesis.



(a) Statistic of the question: "How often did you use a Virtual Reality Headset in the last six month?". As can be seen, 19 people haven't used a VR headset in the last month.

(b) Statistic showing which headset was used by the subjects who doesn't selected the option "Never" in the question covering the use of a VR headset. The bar chart shows, that 60 % of those people used the HTC Vive.

FIGURE A.7: Statistics regarding the users' experience with VR headsets.



(a) Statistic of the question: "Did you know the Myo armband before this experiment?". As the bar chart shows, only nine people knew the Myo and therefrom, only three people have used it themselves.

(b) This bar chart shows how frequently the Myo was used by the subjects who stated, that they used it before. Therefrom, two people used it more than three times a week and one person used it less than three times a month.

FIGURE A.8: Statistics outlining the users' knowledge of the Myo and their experience with it.

A.2 USER STUDY: DIFFERENT NAVIGATION APPROACHES

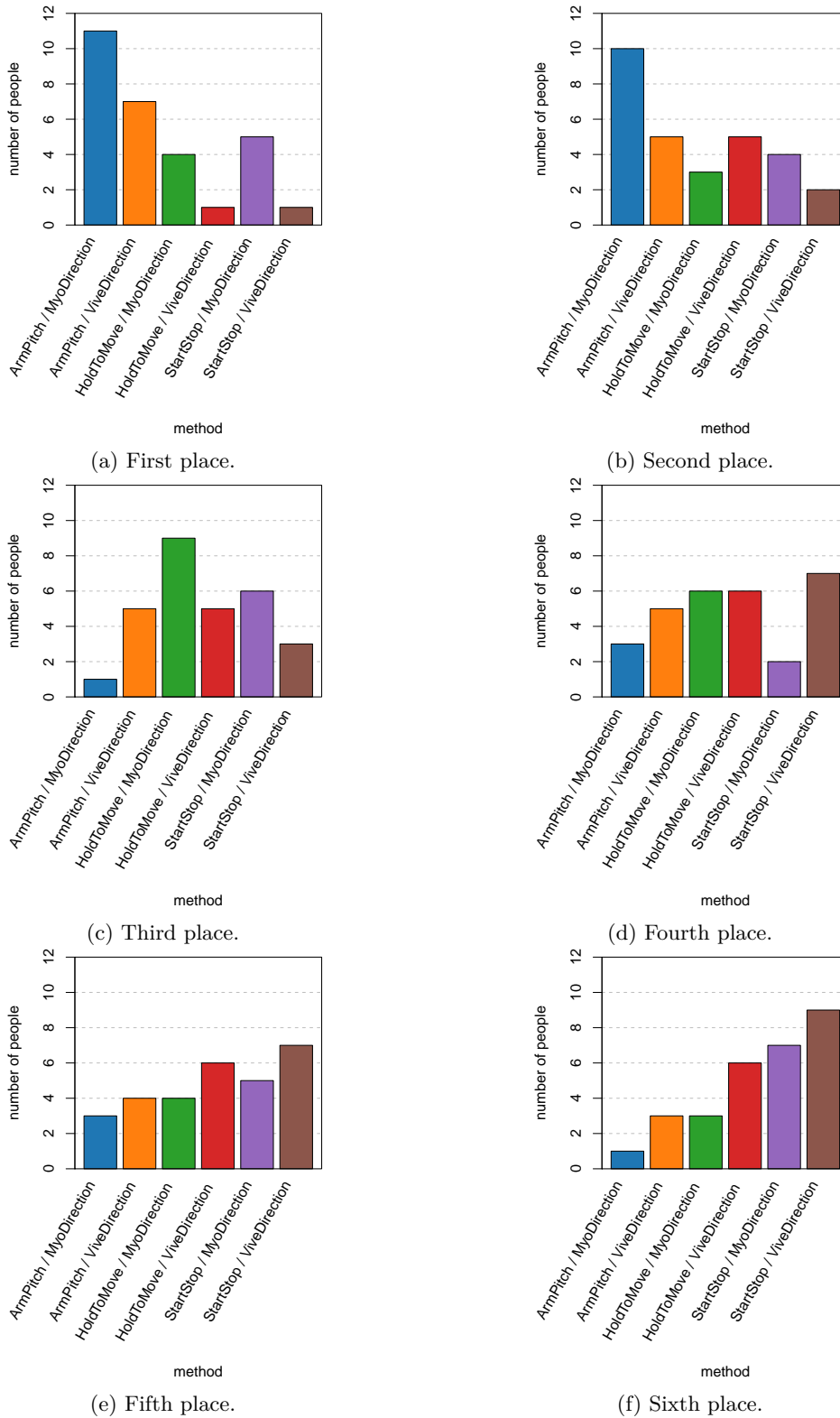


FIGURE A.9: Bar charts showing the users' placements of the different navigation methods.

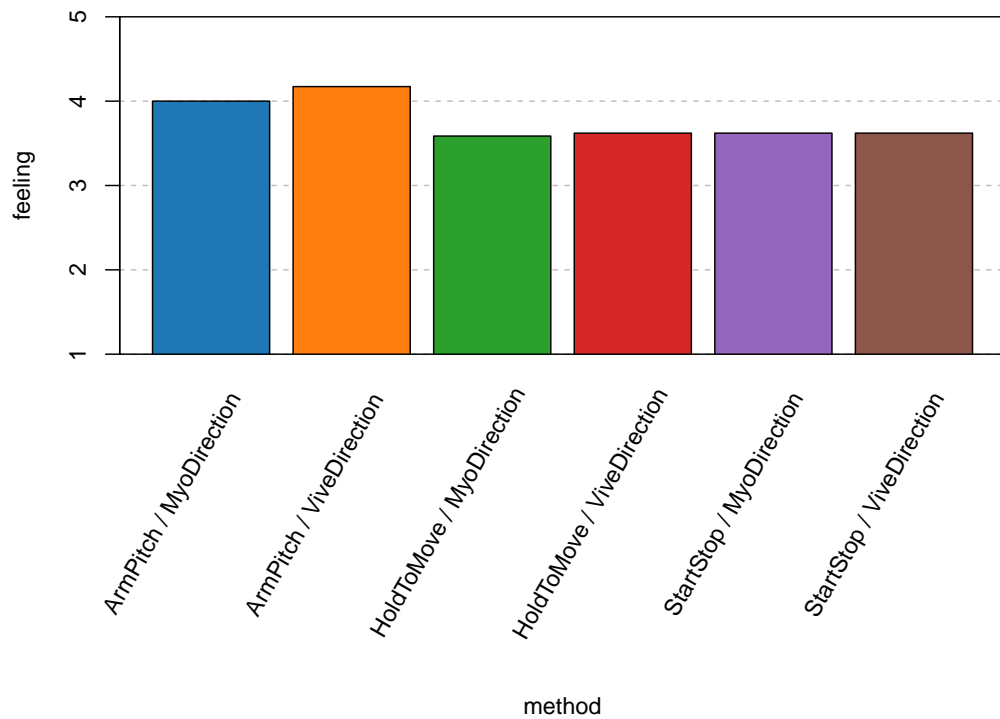


FIGURE A.10: This chart shows the mean rating the users assigned to the different navigation methods in terms of their feeling while utilizing this method. Thereby, 1 indicates a very bad feeling and 5 a very good one.

A.3 USER STUDY: COMPARISON OF GESTURE- AND CONTROLLER-BASED NAVIGATION

A.3 USER STUDY: COMPARISON OF GESTURE- AND CONTROLLER-BASED NAVIGATION

This section includes all relevant materials concerning the second user study, where the re-implemented gesture-based navigation method was compared with a controller-based method.

A.3.1 FIGURES USED TO EXPLAIN THE DIFFERENT NAVIGATION APPROACHES

To explain the different navigation approaches to the user, the following figures were used.

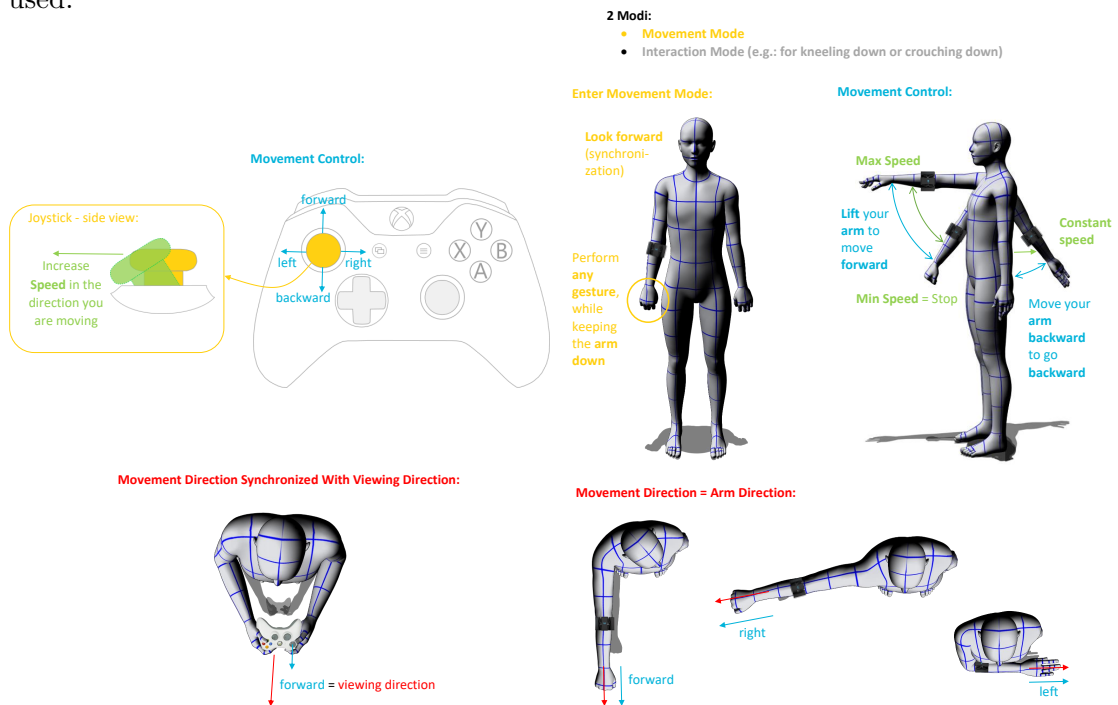


FIGURE A.11: Explanation for the controller-based navigation method

FIGURE A.12: Explanation for the gesture-based navigation method

The blue text describes how the movements can be controlled, whereas, the red text explains in which direction the movement is performed. To illustrate the relationship between movement control and direction, the avatar figures depicted from a top-view are used. Depending on the method, the speed can be either controlled through the forward-pressing of the left joystick, or through lifting the arm as described by the green text. The yellow text is used to highlight the movement activation. In case of the controller-based method, the movement is activated through a push of the left joystick.

The gesture-based method, on the other hand, distinguishes between a movement and an interaction mode. Thereby, the movement mode is activated through an arbitrary gesture of the arm. It is very important to note, that the user needs to look forward during this activation, as the synchronization of the forward direction with the viewing direction happens in this phase.

A.3.2 INFORMATION FOR THE USERS

The following information were given to all users who took place in the user study. As all users have German as their native language, the information was given in German to prevent misunderstandings due to language difficulties.

Summing up the following information, it briefly describes the background and incentive behind this thesis and the user study. In addition to that, the user is provided with information about the procedure of the study and what kind of data are raised in it's course. Lastly, the user is informed about potential risks and what to do in such cases.

Nutzerstudie zum Vergleich von Gesten- und Controller-basierter Navigation in VR

Informationen zur Studie für jeden Probanden

Zielsetzung der Arbeit

Der Titel meiner Masterarbeit lautet „Verbindung von Myo und HTC Vive für die Navigation in der Neurorobotics Plattform“. Die Myo ist ein myoelektrisches Armband zur Gestensteuerung und die HTC Vive ist ein Virtual Reality Headset. Die Arbeit wird im Rahmen der Neurorobotics Plattform erstellt. Diese ist ein Unterprojekt des Human Brain Projects, welches darauf abzielt, dass menschliche Gehirn besser zu verstehen. Die Neurorobotics Plattform bietet dabei eine Simulationsumgebung für Roboter, welche mit virtuellen Gehirn-Modellen verknüpft werden. Da diese Roboter später auch in der realen Welt eingesetzt werden sollen, muss die Mensch-Roboter Interaktion gut funktionieren. Um diese simulieren und analysieren zu können, sollen die Menschen per Virtual Reality in die Simulation integriert werden. Damit die Interaktion so realistisch wie möglich ist, muss der Mensch in der Lage sein, sich möglichst einfach und intuitiv im virtuellen Raum bewegen zu können. Um das zu ermöglichen, wurde im Rahmen der Arbeit eine Navigationsmethode entwickelt, welche in dieser Studie mit einem controllerbasierten Navigationsansatz verglichen werden soll.

Ablauf

- Die User Studie besteht aus zwei Teilen welche jeweils für beide Navigationsmethoden durchlaufen werden:
 1. Der erste Teil besteht aus drei unterschiedlichen Leveln welche nacheinander durchlaufen werden. Das erste Level dient dabei als Eingewöhnungsphase.
 2. Der zweite Teil ist ein online Fragebogen, in welchem die Methode bewertet werden soll.
- Die Studie sollte zwischen 20 und 35 Minuten dauern und ist folgendermaßen aufgebaut:
- Als erstes muss sowohl die Myo als auch die HTC Vive eingestellt werden:
 - Bei der Myo wird die Größe des Armbandes eingestellt und das Gerät mit dem Computer synchronisiert
 - Bei der Vive muss der Riemen am Kopf richtig eingestellt werden.
- Danach werden die drei, im ersten Teil zu durchlaufenden, Level kurz anhand eines Videos vorgestellt. Dabei wird erwähnt, dass das erste Level zur Eingewöhnung dient.
- Im Anschluss wird mit dem ersten Teil des Fragebogens begonnen. Dieser besteht aus allgemeinen Fragen zu Alter, Geschlecht und Vorerfahrung. Diese Daten sind für die spätere Evaluierung nötig und werden über einen vom Probanden selbst generierten Code anonymisiert.
- Danach wird mit der ersten Navigationsmethode begonnen. Diese wird zufällig vom System vorgeschlagen um potentiellen Bias durch die Versuchsleitung zu vermeiden.
 - Die Methode wird zuerst anhand einer Skizze kurz vorgestellt. Danach setzt man die HTC Vive auf und es folgen die drei zuvor gezeigten Level
 - Der Proband wird gebeten, sich während dem 2. und 3. Level nicht die Umgebung anzusehen, sondern so schnell wie möglich zum Roboter zu gelangen.
 - Sobald der Roboter erreicht ist wird der Bildschirm blau und die Versuchsleitung wird das nächste Level am Server startet. Dies kann zu kurzen Wartezeiten führen.
 - Nach dem 3. Level kann die HTC Vive abgesetzt werden.

- Im Anschluss kommt der Teil des Fragebogens welcher Informationen zu dieser Navigationsmethode sammelt.
- Danach folgt die nächste Navigationsmethode, welche wieder vorgestellt wird, danach folgen die drei Level und direkt danach kommt wieder der Fragebogen.
- Nach den Fragen zur letzten Navigationsmethode folgen noch drei weitere Fragen welche die Methoden vergleichen.
- Danach ist die Studie beendet.

Datenverarbeitung und -erhebung

In der Studie werden folgende Daten erhoben:

- Zeiten
- Videomaterial von der HTC Vive

Alle Daten die während der Studie erhoben werden, werden anonymisiert und im Rahmen dieser Studie ausgewertet. Um die Korrektheit des Ergebnisses nachprüfbar zu machen, werden die anonymisierten Daten im Rahmen der Masterarbeit veröffentlicht. Erhobenes Videomaterial von der HTC Vive wird nur von mir, Tamara Barounig, persönlich ausgewertet und nicht an dritte weitergegeben oder gezeigt. Der Proband erklärt sich damit durch eine Unterschrift auf der Einverständniserklärung einverstanden.

Risiken

Im Laufe der Studie kann es im praktischen Teil zu Schwindelanfällen oder Übelkeit, sogenannter „Motion Sickness“ kommen. Der Proband möge hier daran erinnert sein, dass er die Studie nach eigenem Ermessen jederzeit abbrechen kann und sich auf keinen Fall Gefahren aussetzen sollte.

Zusätzliche Informationen

Zum Schluss soll noch einmal verdeutlicht werden, dass zu jedem Zeitpunkt der Studie lediglich die Methode und nicht der Proband selbst evaluiert wird. Jedwede Probleme oder Fehler die auftreten sind alleine durch die Methode und nicht durch den Probanden selbst verursacht.

A.3.3 QUESTIONNAIRE

The survey was conducted with a self-hosted LimeSurvey instance and the following questionnaire is the printable version of the survey the users were asked to complete after completing the three levels for each navigation method.

A.3 USER STUDY: COMPARISON OF GESTURE- AND CONTROLLER-BASED NAVIGATION

Thank you for taking part in this Study!

Section A: Preliminary Questions

A1.
Please fill in your self generated identifier. The identifier is a 7 digit code which looks as follows: NFMNBA
NF: The first and second letter of your father's first name in capital letters MM: The month (01 - 12) in which your mother was born NB: The first and second letter of your birthplace in capital letters A: Does your own LASTname begin with a letter in the first (1) half of the alphabet (A - M) or in the second (2) half of the alphabet (N - Z). Please write 1 or 2 respectively.

A2. What is your age?

17 or younger
18-21 years old
22-25 years old
26-29 years old
30 or older

A3. To which gender identity do you most identify?

Female
Male
Other

Other

A4. What is the highest degree or level of school you have completed?
If you're currently enrolled in school, please indicate the highest degree you have received.

High school degree or equivalent
Bachelor's degree (BA)
Diploma's degree (Dipl.)
Master's degree (MA)
Doctorate (PhD)

A5. What is your main discipline?
If you are a student, please select your current subject. If you are employed, please select your working area.

Informatics
Mathematics
Other

A6. How often did you use a Virtual Reality Headset in the last six month?

Never
Less than one to three times a month
One to three times a month
One to three times a week
More than three times a week

A7. Was the headset you used the HTC Vive?

Yes
No

A8. Did you know the Myo armband before this experiment?

No
Yes, I heard from it
Yes, I used it myself

A9. How often did you use the Myo?

Less than one to three times a month
One to three times a month
One to three times a week
More than three times a week

A10. How often do you play games with a controller?

I don't play games
I play games, but never with a controller
Less than one to three times a month
One to three times a month
One to three times a week
More than three times a week

Section B: Practical Part 1/2

B1. Did you experience any difficulties? If yes, to which of the following categories would you assign your difficulties or problems?

No difficulties
Starting and stopping the movement
Defining the movement direction
Defining the speed
Looking around
Precision of movement

B2. Could you please explain the difficulties or problems in more detail?

B3. How did the proposed method made you feel?
(1 = very bad; 5 = very good)

1
2
3
4
5

B4. How precise were you able to navigate with this method?
(1 = very imprecise; 5 = very precise)

1
2
3
4
5

B5. How dizzy or motion sick let the method feel you?
(1 = not dizzy or motion sick; 5 = very dizzy and motion sick)

1
2
3
4
5

B6. How much pain did you feel during the usage of the method?
(1 = no pain; 5 = a lot of pain)

1
2
3
4
5

CHAPTER A: APPENDIX

B7. Please specify where you feel pain.

B8. Any comments to the presented navigation technique?

Section C: Practical Part 2/2

C1. Did you experience any difficulties? If yes, to which of the following categories would you assign your difficulties or problems?

- No difficulties
- Starting and stopping the movement
- Defining the movement direction
- Defining the speed
- Looking around
- Precision of movement

C2. Could you please explain the difficulties or problems in more detail?

C7. Please specify where you feel pain.

C8. Any comments to the presented navigation technique?

Section D: Final Questions

D1. Which navigation method do you prefer?

- Controller-based navigation (Xbox Controller)
- Gesture-based navigation (Myo)

D2. Why do you prefer the selected method over the other?

D3. Please provide any critical or positive feedback if you like, as both is very valuable for the study.

Thank you very much! You really saved me and my Master Thesis!

C3. How did the proposed method made you feel?

(1 = very bad; 5 = very good)

- 1
- 2
- 3
- 4
- 5

C4. How precise were you able to navigate with this method?

(1 = very imprecise; 5 = very precise)

- 1
- 2
- 3
- 4
- 5

C5. How dizzy or motion sick let the method feel you?

(1 = not dizzy or motion sick; 5 = very dizzy and motion sick)

- 1
- 2
- 3
- 4
- 5

C6. How much pain did you feel during the usage of the method?

(1 = no pain; 5 = a lot of pain)

- 1
- 2
- 3
- 4
- 5

A.3 USER STUDY: COMPARISON OF GESTURE- AND CONTROLLER-BASED NAVIGATION

A.3.4 DATA GATHERED DURING THE PRACTICAL PART

During the practical part of the user study, the order at which the different navigation methods were tested and the corresponding time needed to complete the stage were saved in a CSV file. The resulting raw data is listed in the following table.

Identifier	Order	Stage	Navigation Method	Time [s]
AL04ST2	0	OpenAre	Controller	81.222
AL04ST2	1	Pioneer	Controller	16.856
AL04ST2	2	Husky	Controller	73.244
AL04ST2	3	OpenAre	Myo	120.367
AL04ST2	4	Pioneer	Myo	16.278
AL04ST2	5	Husky	Myo	51.200
AL12FR2	0	OpenAre	Myo	99.233
AL12FR2	1	Pioneer	Myo	20.056
AL12FR2	2	Husky	Myo	81.700
AL12FR2	3	OpenAre	Controller	58.633
AL12FR2	4	Pioneer	Controller	18.733
AL12FR2	5	Husky	Controller	32.456
FA09RO2	0	OpenAre	Controller	53.300
FA09RO2	1	Pioneer	Controller	13.356
FA09RO2	2	Husky	Controller	31.889
FA09RO2	3	OpenAre	Myo	39.622
FA09RO2	4	Pioneer	Myo	13.489
FA09RO2	5	Husky	Myo	48.456
FR10PR1	0	OpenAre	Controller	24.989
FR10PR1	1	Pioneer	Controller	11.811
FR10PR1	2	Husky	Controller	31.711
FR10PR1	3	OpenAre	Myo	22.867
FR10PR1	4	Pioneer	Myo	20.311
FR10PR1	5	Husky	Myo	175.478
FR11BE1	0	OpenAre	Myo	69.656
FR11BE1	1	Pioneer	Myo	23.078
FR11BE1	2	Husky	Myo	46.778
FR11BE1	3	OpenAre	Controller	11.467
FR11BE1	4	Pioneer	Controller	13.589
FR11BE1	5	Husky	Controller	31.656
FR11MU2	0	OpenAre	Myo	58.089
FR11MU2	1	Pioneer	Myo	15.022
FR11MU2	2	Husky	Myo	98.933
FR11MU2	3	OpenAre	Controller	18.789
FR11MU2	4	Pioneer	Controller	13.456
FR11MU2	5	Husky	Controller	26.833
FZ06ER2	0	OpenAre	Myo	29.556
FZ06ER2	1	Pioneer	Myo	47.411
FZ06ER2	2	Husky	Myo	196.367
FZ06ER2	3	OpenAre	Controller	34.033
FZ06ER2	4	Pioneer	Controller	41.822
FZ06ER2	5	Husky	Controller	143.756

CHAPTER A: APPENDIX

Identifier	Order	Stage	Navigation Method	Time [s]
GE01MU1	0	OpenAre	Controller	38.789
GE01MU1	1	Pioneer	Controller	13.622
GE01MU1	2	Husky	Controller	22.233
GE01MU1	3	OpenAre	Myo	35.611
GE01MU1	4	Pioneer	Myo	11.967
GE01MU1	5	Husky	Myo	59.267
GE01RE1	0	OpenAre	Controller	43.522
GE01RE1	1	Pioneer	Controller	11.889
GE01RE1	2	Husky	Controller	67.378
GE01RE1	3	OpenAre	Myo	30.600
GE01RE1	4	Pioneer	Myo	17.822
GE01RE1	5	Husky	Myo	47.478
GE05MA1	0	OpenAre	Myo	25.800
GE05MA1	1	Pioneer	Myo	44.189
GE05MA1	2	Husky	Myo	140.500
GE05MA1	3	OpenAre	Controller	4.533
GE05MA1	4	Pioneer	Controller	23.422
GE05MA1	5	Husky	Controller	66.611
GE11PA2	0	OpenAre	Myo	84.022
GE11PA2	1	Pioneer	Myo	32.578
GE11PA2	2	Husky	Myo	56.022
GE11PA2	3	OpenAre	Controller	19.556
GE11PA2	4	Pioneer	Controller	14.022
GE11PA2	5	Husky	Controller	30.633
GU06MU1	0	OpenAre	Controller	38.756
GU06MU1	1	Pioneer	Controller	30.356
GU06MU1	2	Husky	Controller	56.600
GU06MU1	3	OpenAre	Myo	23.422
GU06MU1	4	Pioneer	Myo	28.100
GU06MU1	5	Husky	Myo	55.322
HE02MU2	0	OpenAre	Myo	36.778
HE02MU2	1	Pioneer	Myo	25.167
HE02MU2	2	Husky	Myo	224.411
HE02MU2	3	OpenAre	Controller	16.944
HE02MU2	4	Pioneer	Controller	16.089
HE02MU2	5	Husky	Controller	37.522
HE08MU1	0	OpenAre	Controller	31.644
HE08MU1	1	Pioneer	Controller	22.133
HE08MU1	2	Husky	Controller	43.200
HE08MU1	3	OpenAre	Myo	31.278
HE08MU1	4	Pioneer	Myo	17.956
HE08MU1	5	Husky	Myo	89.411
HU07BI2	0	OpenAre	Myo	45.289
HU07BI2	1	Pioneer	Myo	14.911
HU07BI2	2	Husky	Myo	85.944
HU07BI2	3	OpenAre	Controller	7.422
HU07BI2	4	Pioneer	Controller	11.700
HU07BI2	5	Husky	Controller	19.367

A.3 USER STUDY: COMPARISON OF GESTURE- AND CONTROLLER-BASED NAVIGATION

Identifier	Order	Stage	Navigation Method	Time [s]
JO06GR1	0	OpenAre	Myo	53.544
JO06GR1	1	Pioneer	Myo	24.889
JO06GR1	2	Husky	Myo	57.378
JO06GR1	3	OpenAre	Controller	28.544
JO06GR1	4	Pioneer	Controller	23.800
JO06GR1	5	Husky	Controller	45.767
JO08CO2	0	OpenAre	Controller	10.378
JO08CO2	1	Pioneer	Controller	11.133
JO08CO2	2	Husky	Controller	59.722
JO08CO2	3	OpenAre	Myo	11.389
JO08CO2	4	Pioneer	Myo	13.044
JO08CO2	5	Husky	Myo	46.833
KL01IN2	0	OpenAre	Controller	25.367
KL01IN2	1	Pioneer	Controller	21.467
KL01IN2	2	Husky	Controller	84.244
KL01IN2	3	OpenAre	Myo	40.611
KL01IN2	4	Pioneer	Myo	32.889
KL01IN2	5	Husky	Myo	46.156
LU12SP2	0	OpenAre	Controller	20.600
LU12SP2	1	Pioneer	Controller	24.433
LU12SP2	2	Husky	Controller	50.167
LU12SP2	3	OpenAre	Myo	11.833
LU12SP2	4	Pioneer	Myo	18.422
LU12SP2	5	Husky	Myo	55.322
MI07WO1	0	OpenAre	Controller	20.911
MI07WO1	1	Pioneer	Controller	18.144
MI07WO1	2	Husky	Controller	42.056
MI07WO1	3	OpenAre	Myo	5.311
MI07WO1	4	Pioneer	Myo	25.567
MI07WO1	5	Husky	Myo	44.833
MI09MU1	0	OpenAre	Myo	13.300
MI09MU1	1	Pioneer	Myo	27.833
MI09MU1	2	Husky	Myo	47.411
MI09MU1	3	OpenAre	Controller	9.133
MI09MU1	4	Pioneer	Controller	12.356
MI09MU1	5	Husky	Controller	58.767
OT12WA1	0	OpenAre	Controller	63.789
OT12WA1	1	Pioneer	Controller	17.722
OT12WA1	2	Husky	Controller	44.111
OT12WA1	3	OpenAre	Myo	26.678
OT12WA1	4	Pioneer	Myo	21.433
OT12WA1	5	Husky	Myo	74.422
PE11MU1	0	OpenAre	Controller	49.811
PE11MU1	1	Pioneer	Controller	12.344
PE11MU1	2	Husky	Controller	27.811
PE11MU1	3	OpenAre	Myo	33.067
PE11MU1	4	Pioneer	Myo	12.822
PE11MU1	5	Husky	Myo	31.833

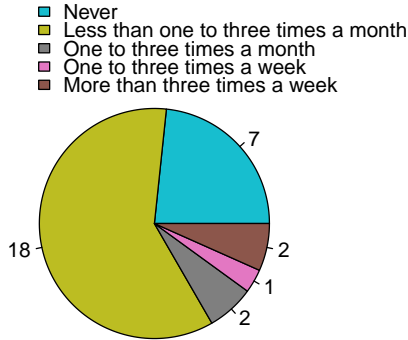
CHAPTER A: APPENDIX

Identifier	Order	Stage	Navigation Method	Time [s]
RA07FR1	0	OpenAre	Myo	42.144
RA07FR1	1	Pioneer	Myo	25.256
RA07FR1	2	Husky	Myo	72.911
RA07FR1	3	OpenAre	Controller	20.267
RA07FR1	4	Pioneer	Controller	10.911
RA07FR1	5	Husky	Controller	85.656
RO06LI2	0	OpenAre	Myo	50.444
RO06LI2	1	Pioneer	Myo	34.133
RO06LI2	2	Husky	Myo	54.322
RO06LI2	3	OpenAre	Controller	24.911
RO06LI2	4	Pioneer	Controller	28.900
RO06LI2	5	Husky	Controller	57.089
RO09DA1	0	OpenAre	Controller	11.300
RO09DA1	1	Pioneer	Controller	17.300
RO09DA1	2	Husky	Controller	31.689
RO09DA1	3	OpenAre	Myo	85.589
RO09DA1	4	Pioneer	Myo	22.800
RO09DA1	5	Husky	Myo	62.956
SH12BO2	0	OpenAre	Myo	83.333
SH12BO2	1	Pioneer	Myo	36.078
SH12BO2	2	Husky	Myo	165.656
SH12BO2	3	OpenAre	Controller	31.933
SH12BO2	4	Pioneer	Controller	21.222
SH12BO2	5	Husky	Controller	84.522
TH01MU1	0	OpenAre	Myo	64.433
TH01MU1	1	Pioneer	Myo	32.589
TH01MU1	2	Husky	Myo	70.300
TH01MU1	3	OpenAre	Controller	30.178
TH01MU1	4	Pioneer	Controller	17.522
TH01MU1	5	Husky	Controller	53.289
WE06AU1	0	OpenAre	Myo	57.344
WE06AU1	1	Pioneer	Myo	27.067
WE06AU1	2	Husky	Myo	61.644
WE06AU1	3	OpenAre	Controller	45.111
WE06AU1	4	Pioneer	Controller	25.744
WE06AU1	5	Husky	Controller	46.311
WO04AU2	0	OpenAre	Controller	17.422
WO04AU2	1	Pioneer	Controller	12.167
WO04AU2	2	Husky	Controller	50.189
WO04AU2	3	OpenAre	Myo	41.256
WO04AU2	4	Pioneer	Myo	24.689
WO04AU2	5	Husky	Myo	82.244

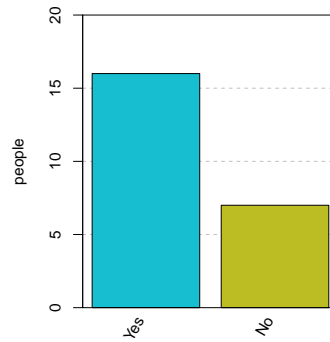
A.3 USER STUDY: COMPARISON OF GESTURE- AND CONTROLLER-BASED NAVIGATION

A.3.5 DETAILED SURVEY RESULTS

This section includes additional survey results not addressed in the main part of this thesis.

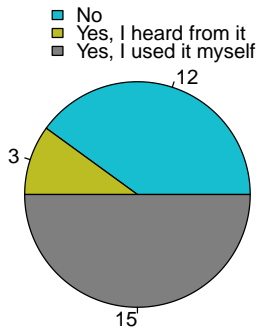


(a) Statistic of the question: "How often did you use a Virtual Reality Headset in the last six month?". As can be seen, only seven people haven't used a VR headset in the last six month and 18 people have used it less than one to three times a month.

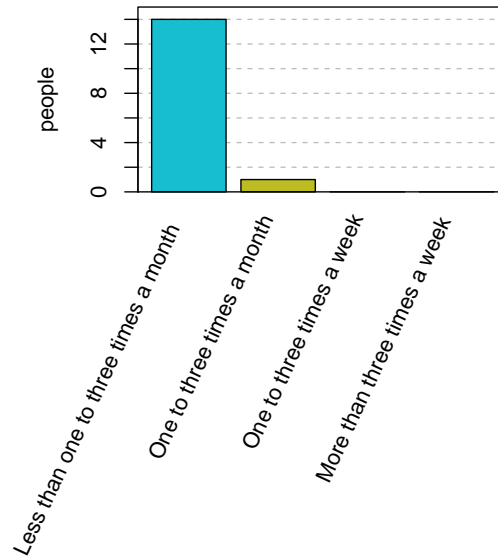


(b) Statistic showing which headset was used by the subjects who doesn't selected the option "Never" in the question covering the use of a VR headset. The bar chart shows, that 69.6 % of those people used the HTC Vive.

FIGURE A.13: Statistics regarding the users' experience with VR headsets.



(a) Statistic of the question: "Did you know the Myo armband before this experiment?". As the bar chart shows, half of the people already used the Myo and only 12 people did not knew the Myo at all.



(b) This bar chart shows how frequently the Myo was used by the subjects who stated, that they used it before. Therefrom, only one person used it more than three times a month.

FIGURE A.14: Statistics outlining the users' knowledge of the Myo and their experience with it.

CHAPTER B

LIST OF ACRONYMS

CSV	Comma-Separated Values
DART	Dynamic Animation and Robotics Toolkit
EMG	Electromyography
HBP	Human Brain Project
HCI	Human-Computer Interaction
HMD	Head Mounted Display
ICT	Information and Communications Technology
IMU	Inertial Measurement Unit
NRP	Neurorobotics Platform
ODE	Open Dynamics Engine
ROS	Robot Operating System
SDF	Simulation Description Format
SNN	Spiking Neural Network
TUM	Technical University of Munich
VR	Virtual Reality

BIBLIOGRAPHY

- [Bis92] Bishop, Gary and Bricken, William and Brooks, Frederick and Brown, Marcus and Burbeck, Chris and Durlach, Nat and Ellis, Steve and Fuchs, Henry and Green, Mark and Lackner, James and McNeill, Michael and Moshell, Michael and Pausch, Randy and Robinett, Warren and Srinivasan, Mandayam and Sutherland, Ivan and Urban, Dick and Wenzel, Elizabeth. *Research Directions in Virtual Environments - Report of an NSF Invitational Workshop March 23-24, 1992*. University of North Carolina at Chapel Hill, 1992.
- [Bon17] Bonome, Yadira Garnica. “Proposing two new handling interaction techniques for 3D virtual objects using the Myo armband”. <https://www.maxwell.vrac.puc-rio.br/30744/30744.PDF>. MA thesis. Rio de Janeiro: Pontifical Catholic University of Rio de Janeiro, May 2017.
- [Bur03] Burdea, Grigore C. and Coiffet, Philippe. *Virtual Reality Technology*. Second Edition. New Jersey: John Wiley & Sons, Inc., 2003.
- [Cab05] Cabral, Marcio C. and Morimoto, Carlos H. and Zuffo, Marcelo K. “On the Usability of Gesture Interfaces in Virtual Reality Environments”. In: *Proceedings of the 2005 Latin American Conference on Human-computer Interaction*. CLIHC '05. Cuernavaca, Mexico: ACM, 2005, 100–108. DOI: {10.1145/1111360.1111370}.
- [Cod14] Codd-Downey, Robert and Stuerzlinger, Wolfgang. “LeapLook: A Free-hand Gestural Travel Technique Using the Leap Motion Finger Tracker”. In: *Proceedings of the 2Nd ACM Symposium on Spatial User Interaction*. SUI '14. Honolulu, Hawaii, USA: ACM, 2014, 153–153. DOI: {10.1145/2659766.2661218}.
- [Eurnda] European Brain Council. *Human Brain Project*. <https://www.braincouncil.eu/tag/understanding-the-brain/>. Accessed: 2017-11-03. n.d.

BIBLIOGRAPHY

- [Eurndb] European Commission. *FET Flagships*. <http://ec.europa.eu/programmes/horizon2020/en/h2020-section/fet-flagships>. Accessed: 2017-11-03. n.d.
- [Fal17a] Falotico, Egidio. *A comprehensive framework for connecting simulated robots to artificial brains: The Neurorobotics Platform*. https://www.eu-robotics.net/robotics_forum/upload/events/1055/files/1055_Falotico_The-NRP.pdf. Accessed: 2017-11-13. 2017.
- [Fal17b] Falotico, Egidio and Vannucci, Lorenzo and Ambrosano, Alessandro and Albanese, Ugo and Ulbrich, Stefan and Vasquez Tieck, Juan Camilo and Hinkel, Georg and Kaiser, Jacques and Peric, Igor and Denninger, Oliver and Cauli, Nino and Kirtay, Murat and Roennau, Arne and Klinker, Gudrun and Von Arnim, Axel and Guyot, Luc and Peppicelli, Daniel and Martínez-Cañada, Pablo and Ros, Eduardo and Maier, Patrick and Weber, Sandro and Huber, Manuel and Plecher, David and Röhrbein, Florian and Deser, Stefan and Roitberg, Alina and van der Smagt, Patrick and Dillman, Rüdiger and Levi, Paul and Laschi, Cecilia and Knoll, Alois C. and Gewaltig, Marc-Oliver. “Connecting Artificial Brains to Robots in a Comprehensive Simulation Framework: The Neurorobotics Platform”. In: *Frontiers in Neurorobotics* Volume 11 (2017), 2. ISSN: 1662-5218.
- [For14] Forouher, Dariush. *Topics*. <http://wiki.ros.org/Topics>. Accessed: 2017-11-17. 2014.
- [Gaznda] Gazebo.org. *Gazebo Architecture*. <http://gazebo.org/tutorials?tut=architecture>. Accessed: 2017-11-17. n.d.
- [Gazndb] Gazebo.org. *Gazebo Components*. http://gazebo.org/tutorials?tut=components&cat=get_started. Accessed: 2017-11-17. n.d.
- [Gazndc] Gazebo.org. *Why Gazebo*. <http://gazebo.org/>. Accessed: 2017-11-17. n.d.
- [HTCnd] HTC Corporation. *Vive - Products*. <https://www.vive.com/uk/product/>. Accessed: 2018-01-31. n.d.
- [Hum17a] Human Brain Project. *Edit the simulation*. https://developer.humanbrainproject.eu/docs/projects/HBP%20Neurorobotics%20Platform/1.2/nrp/user_manual/user_interface/edit/7-gz3d-edit-simulation.html#brain. Accessed: 2017-11-03. 2017.
- [Hum17b] Human Brain Project. *Introduction*. <https://www.humanbrainproject.eu/en/about/project-structure/subprojects/>. Accessed: 2017-10-03. 2017.
- [Hum98] Hummels, Caroline and Stappers, Pieter J. “Meaningful gestures for human computer interaction: beyond hand postures”. In: *Proceedings Third IEEE*

- International Conference on Automatic Face and Gesture Recognition*. Apr. 1998, 591–596. DOI: {10.1109/AFGR.1998.671012}.
- [Humnda] Human Brain Project. *Overview*. <https://www.humanbrainproject.eu/en/about/overview/>. Accessed: 2017-11-03. n.d.
- [Humndb] Human Brain Project. *Subprojects*. <https://www.humanbrainproject.eu/en/about/project-structure/subprojects/>. Accessed: 2017-11-03. n.d.
- [Kim00] Kim, Jong-Soo and Park, Kyu Hyung and Kim, Jung-Bae and Do, Jun-Hyeong and Song, Kyung-Joon and Bien, Zeungnam. “Study on Intelligent Autonomous Navigation of Avatar using Hand Gesture Recognition”. In: *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*. Vol. 2. Oct. 2000, 846–851. DOI: {10.1109/ICSMC.2000.885955}.
- [Koe04] Koenig, Nathan and Howard, Andrew. “Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Sendai, Japan, Sept. 2004, 2149–2154.
- [Kur16] Kurzaj, Dorian. *Messages*. <http://wiki.ros.org/Messages>. Accessed: 2017-11-17. 2016.
- [Lee98] Lee, Chan-Su and Ghyme, Sang-Won and Park, Chan-Jong and Wohn, Kwangyun. “The Control of Avatar Motion Using Hand Gesture”. In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. VRST '98. Taipei, Taiwan: ACM, 1998, 59–65. DOI: {10.1145/293701.293709}.
- [Limnd] LimeSurvey GmbH. *Professional online surveys with LimeSurvey*. <https://www.limesurvey.org/>. Accessed: 2018-03-20. n.d.
- [Lud16] Ludwig, Joe. *Using Unity at Valve*. <https://www.youtube.com/watch?v=4Gs5k2Fti1U>. Channel: Vision VR/AR Summit. Accessed: 2018-03-16. Feb. 2016.
- [Luh15] Luh, Guan-Chun and Lin, Heng-An and Ma, Yi-Hsiang and Yen, Chien Jung. “Intuitive muscle-gesture based robot navigation control using wearable gesture armband”. In: *2015 International Conference on Machine Learning and Cybernetics (ICMLC)*. Vol. 1. July 2015, 389–395. DOI: {10.1109/ICMLC.2015.7340953}.
- [Mar14a] Mark. *GUI without the G: Going Beyond the Screen with the Myo Armband*. <http://developerblog.myo.com/gui-without-g-going-beyond-screen-myotm-armband/>. Accessed: 2018-03-12. 2014.
- [Mar14b] Martinez Romero, Aaron. *ROS/Concepts*. <http://wiki.ros.org/ROS/Concepts>. Accessed: 2017-11-16. 2014.

BIBLIOGRAPHY

- [Maz99] Mazuryk, Tomasz and Gervautz, Michael. *Virtual Reality - History, Applications, Technology and Future*. Institute of Computer Graphics, Vienna University of Technology, Austria, Dec. 1999.
- [McC15] McCullough, Morgan and Xu, Hong and Michelson, Joel and Jackoski, Matthew and Pease, Wyatt and Cobb, William and Kalescky, William and Ladd, Joshua and Williams, Betsy. “Myo Arm-Swinging to explore a VE”. In: *SAP '15 Proceedings of the ACM SIGGRAPH Symposium on Applied Perception*. Tübingen, Germany: ACM, Sept. 2015, 107–113. DOI: {10.1145/2804408.2804416}.
- [Micnd] Microsoft. *Build Unity Games with Visual Studio*. <https://www.visualstudio.com/vs/unity-tools/>. Accessed: 2018-03-16. n.d.
- [Neund] Neurorobotics.net. *Neurorobotics Platform - Brains for bodies and bodies for brains*. <https://neurorobotics.net/>. Accessed: 2017-10-03. n.d.
- [Nie04] Nielsen, Michael and Störring, Moritz and Moeslund, Thomas B. and Granum, Erik. “A Procedure for Developing Intuitive and Ergonomic Gesture Interfaces for HCI”. In: *Gesture-Based Communication in Human-Computer Interaction*. Ed. by Camurri, Antonio and Volpe, Gualtiero. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, 409–420.
- [Pat15] Patel, Jayam Umesh. “Gesture Based Navigation and Localization of a Smart Wheelchair using Fiducial Markers”. <https://web.wpi.edu/Pubs/ETD/Available/etd-042816-031040/unrestricted/jpatel.pdf>. MA thesis. Worcester, Massachusetts, USA: Worcester Polytechnic Institute, Apr. 2015.
- [Pl015] Plouzeau, Jérémy and Paillot, Damien and Chardonnet, Jean-Rémy and Merienne, Frédéric. “Effect of proprioceptive vibrations on simulator sickness during navigation task in virtual environment”. In: *International Conference on Artificial Reality and Telexistence Eurographics Symposium on Virtual Environments*. hal-01230568. Kyoto, Japan, Oct. 2015, 1–6.
- [Rea06] Reaz, M. B. I. and Hussain, M. S. and Mohd-Yasin, F. “Techniques of EMG signal analysis: detection, processing, classification and applications”. In: *Biological Procedures Online* 8.1 (Dec. 2006), 11–35. ISSN: 1480-9222. DOI: {10.1251/bpo115}.
- [Rie05] Riecke, Bernhard E. and Schulte-Pelkum, Jörg and Caniard, Franck and Bulthoff, Heinrich H. “Towards lean and elegant self-motion simulation in virtual reality”. In: *IEEE Proceedings. VR 2005. Virtual Reality, 2005*. Mar. 2005, 131–138. DOI: {10.1109/VR.2005.1492765}.
- [Rie10] Riecke, Bernhard E. and Bodenheimer, Bobby and McNamara, Timothy P. and Williams, Betsy and Peng, Peng and Feuereissen, Daniel. “Do We

- Need to Walk for Effective Virtual Reality Navigation? Physical Rotations Alone May Suffice”. In: *Spatial Cognition VII*. Berlin, Heidelberg: Springer Berlin Heidelberg, Aug. 2010, 234–247. ISBN: 978-3-642-14749-4.
- [Rosnda] Ros.org. *About ROS*. <http://www.ros.org/about-ros/>. Accessed: 2017-11-14. n.d.
- [Rosndb] Ros.org. *Integration with Other Libraries*. <http://www.ros.org/integration/>. Accessed: 2017-11-17. n.d.
- [Rö16] Röhrbein, Florian. *The Human Brain Project: Prototype of the Neuro-robotics Platform Released - Virtual Robots for the Simulated Brain*. <https://www.tum.de/en/about-tum/news/press-releases/detail/article/33022/>. Accessed: 2017-11-03. 2016.
- [She03] Sherman, William R. and Craig, Alan B. *Understanding Virtual Reality: Interface, Application, and Design*. San Francisco: Morgan Kaufmann, 2003.
- [Sut65] Sutherland, Ivan E. “The Ultimate Display”. In: Volume 2 (1965), 506–509.
- [Thanda] Thalnic Labs Inc. *Getting starting with Myo on Windows*. <https://support.getmyo.com/hc/en-us/articles/202657596-Getting-starting-with-Myo-on-Windows>. Accessed: 2018-03-05. n.d.
- [Thandb] Thalnic Labs Inc. *Getting To Know The Myo Armband*. https://developer.thalnic.com/docs/api_reference/platform/getting-started.html. Accessed: 2018-03-05. n.d.
- [Thandc] Thalnic Labs Inc. *Getting To Know The SDK*. https://developer.thalnic.com/docs/api_reference/platform/the-sdk.html. Accessed: 2018-03-15. n.d.
- [Thandd] Thalnic Labs Inc. *Lesson One: Getting Started with Myo*. <https://support.getmyo.com/hc/en-us/articles/115003567723-Myo-Lesson-Plans>. Accessed: 2018-03-05. n.d.
- [Thande] Thalnic Labs Inc. *Using the Armband Manager in Myo Connect*. <https://support.getmyo.com/hc/en-us/articles/205031939-Using-the-Armband-Manager-in-Myo-Connect>. Accessed: 2018-03-16. n.d.
- [Thandf] Thalnic Labs Inc. *What gestures does the Myo armband recognize?* <https://support.getmyo.com/hc/en-us/articles/202647853-What-gestures-does-the-Myo-armband-recognize>. Accessed: 2018-03-05. n.d.
- [Thandg] Thalnic Labs Inc. *What’s included?* <https://www.myo.com/techspecs>. Accessed: 2018-03-05. n.d.
- [Thend] The NEST Initiative. *What is nest?* <http://www.nest-simulator.org/>. Accessed: 2017-11-14. n.d.
- [Tho14] Thomas, Dirk. *ROS/Introduction*. <http://wiki.ros.org/ROS/Introduction>. Accessed: 2017-11-14. 2014.

BIBLIOGRAPHY

- [Von95] Von Schweber, Linda and Erick. “Virtually Here”. In: Volume 14, No. 5 (Mar. 1995), 168–198.
- [Wad16] Wada, Kentaro. *rostopic - Package Summary*. <http://wiki.ros.org/rostopic>. Accessed: 2017-11-17. 2016.
- [Zha17] Zhang, Fan and Chu, Shaowei and Pan, Ruifang and Ji, Naye and Xi, Lian. “Double hand-gesture interaction for walk-through in VR environment”. In: *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*. May 2017, 539–544. DOI: {10.1109/ICIS.2017.7960051}.
- [Şe12] Şen, Ferhat and Díaz, Lily and Horttana, Tommi. “A novel gesture-based interface for a VR simulation: Re-discovering Vrouw Maria”. In: *2012 18th International Conference on Virtual Systems and Multimedia*. Sept. 2012, 323–330. DOI: {10.1109/VSMM.2012.6365941}.