2



Navigation in Virtual Reality Space

Concepts of Navigation Methods

**IP5 Project of**

**Bär Dominic**

**Groux Marcel**

**FHNW**

**University of Applied Sciences**

**Degree Course: Computer Science / iCompetence**

**Supervising Lecturers: Arizona Stefan, Marcin Simon**

**Windisch, 21. January 2017**

Clarification of Honest

Hereby I declare to have written the present IP5 Project independently, without help of a third party and only under the usage of the declared sources.

|  |  |  |
| --- | --- | --- |
| Brugg, 21 January 2017 |  | Brugg, 21 January 2017 |
| Place, date |  | Place, date |
|  |  |  |
| Signature Dominic Bär |  | Signature Marcel Groux |

Summary (Both)

Project Summary

Preface (Both)

Vorwort mit Danksagung

Index

1 Introduction (Dominic) 1

1.1 What has been achieved? 1

1.2 Why has it been done? 1

1.3 How has it been achieved? 1

1.4 Readers Guide: How is the rest of the document constructed? 1

2 Initial Position 2

2.1 Introduction 2

2.2 Application domain 2

2.3 Overall scenario 2

2.3.1 Target audience 2

2.4 Project Goals 2

2.4.1 Navigation Methods 3

2.5 Project Scope 3

2.6 Limitations and Assumptions 3

2.6.1 Limitations 3

2.6.2 Assumptions 3

3 Research 4

3.1 Introduction 4

3.2 Problem 4

3.3 Researched Navigation Methods 4

3.3.1 Implemented Navigation Methods 5

3.3.2 Other Navigation Methods 7

3.3.3 Researched Parameters (MARCEL) 10

3.4 Technical Research 10

3.4.1 Game Engines 11

3.4.2 VR Headsets 12

4 Implementation (Marcel) 14

4.1 Introduction (Dominic) 14

4.2 Walking in Place 14

4.2.1 Concept & Idea 14

4.2.2 Implementation 14

4.2.3 Parameters 14

4.3 Scaled Walking 15

4.3.1 Concept & Idea 15

4.3.2 Implementation 15

4.3.3 Parameters 15

4.4 Walking by Leaning 15

4.4.1 Concept & Ideas 15

4.4.2 Implementation 16

4.4.3 Parameters 16

4.5 Jumping 16

4.5.1 Concept & Idea 16

4.5.2 Implementation 17

4.5.3 Parameters 17

4.6 Combining the navigation methods 17

5 Testing (BOTH) 18

5.1 Introduction (Dominic) 18

5.2 Testing Szenario 18

5.3 Experience with Virtual Reality 18

5.4 Ease of Learning 18

5.4.1 Teleport 19

5.4.2 Jumping 19

5.4.3 Walking in Place 20

5.4.4 Walking by Leaning 20

5.4.5 Comparing all Navigation methods. 20

5.5 Pick & place 21

5.5.1 Teleport 22

5.5.2 Jumping 23

5.5.3 Walking in Place 24

5.5.4 Walking by Leaning 25

5.5.5 Overall 25

5.6 Jump’n’Run 25

5.6.1 Teleport 26

5.6.2 Jumping 28

5.7 Ease of Use 29

5.8 Problems during testing 29

6 Conclusion (BOTH) 30

6.1 Introduction 30

6.2 Insights 30

6.3 Suggestions 30

7 Further Steps 31

7.1 Introduction 31

7.2 Marketplace UE4 / Unity3D 31

7.3 Graphical Navigation Menu / UI 31

7.4 Composition of Navigation methods 31

8 Reflection (Both) 32

8.1 Introduction 32

8.2 Lessons Learned 32

8.2.1 Dominic Bär 32

8.2.2 Marcel Groux 32

8.3 Time Management 33

8.4 Collaboration 34

8.4.1 Team Internal Collaboration 34

8.4.2 Collaboration with Coaches / Clients 34

8.4.3 Collaboration with the ‘Explorative Navigation in Virtual Reality’ project team 34

9 Index of Literature 35

L1. Internet 35

L2. Existing Projects 35

10 Index of Figures 36

10.1 Figures 36

10.2 Chartpairs 36

A. Attachment A1

A1. Project Agreement A1

A2. Test Procedure A3

A3. Testing Survey A6

A4. Testing Survey Results A15

A4.1. Ease of Learning A15

A4.2. Pick & Place A17

A4.3. Jump’n’Run A19

A4.4. Ease of Use A21

1. Introduction (Dominic)

This chapter contains an overview of the project. It describes what has been accomplished with the project and which topics are covered.

## What has been achieved?

Within the scope of this project a prototype for methods of navigation in the virtual reality space has been created. This prototype contains five different methods of navigation, from these 5 we tested four movements methods with participants.

To further use this prototype in upcoming projects a concept containing suggestions for using the different navigation methods has been created.We also discuss further possible enhancements to the prototype.

## Why has it been done?

The prototype was created to analyse the navigation methods in the virtual reality and to create suggestions on which navigation method to use in which environment and / or scenario.

## How has it been achieved?

The creation of the prototype can be divided into two parts. In the first part we researched many different navigation methods and their used parameters. Based on those we chose certain navigation methods to implement and created a concept and idea how we imagined them to be implemented. The second part covers the implementation to create the prototype for the chosen navigation methods.

Regarding the technical aspects, we used the game engine UnrealEngine 4 and the virtual reality device HTC Vive.

## Readers Guide: How is the rest of the document constructed?

The document consists of two parts. The first one contains the theoretical aspects covering the goals and the research. The second part addresses the practical aspects of implementation, testing and suggestions.

# Initial Position

## Introduction

In this chapter the initial position of the project will be introduced. The Application domain will be described and an overall scenario will be shown. Furthermore, the project goals and scope will be stated.

## Application domain

In a first instance the project was created as a proof of concept for the University of Applied Science FHNW and has currently no direct application domain. In a second instance the project could be published and find its usage in the development of virtual reality applications or games.

## Overall scenario

The project covers the research and analyzation of navigation method and the development of a prototype for navigation methods in the virtual reality. The overall goal is to create a concept of different navigation methods with suggestions for each method and their suitability in different scenarios.

### Target audience

The creation of the prototype is targeted for creators of virtual reality applications or games used in a home environment.

## Project Goals

The goal of this project is the generation of a concept about the navigation in the Virtual Reality space. The concept is based on a scientific research and should address the questions of the suitability for different navigation methods and the corresponding parameters (e.g. camera angle/area, scaling in space, …) within specific scenarios, which are to be determined.

Finally, the concept contains a thorough scientific analysis of VR navigation and its parameters, elaborated in a scientific approach and reflecting the current state of research of the Virtual Reality Community as far as possible.

The navigation methods, elaborated in the concept, should be implemented as a template for different scenarios and be tested thoroughly. Such that it can be shown which navigation methods are suited best for different scenarios. Thereby it is to bear in mind that the navigation that we are reviewing should be possible to use in a home-user-environment.

### Navigation Methods

The following navigation methods will be elaborated in the prototype:

* Walking in Place
* Walking by Leaning
* Scaled Walking
* Teleporting
* Jumping

Further details to each navigation method will be given in chapter ‘3.3update? Researched Navigation Methods’.

## Project Scope

Project contains the following emphases:

* Research of navigation methods and their respective parameters
* Creation of a concept of how to implement the navigation methods
* Implementation of the chosen navigation methods
* Testing and analysis of the implemented navigation methods

## Limitations and Assumptions

### Limitations

We researched far more navigation methods than we have had to implement and test. Due to that we had to limit the number of navigation methods we implement in our prototype. We chose the one navigation method that is extensively seen all over vr, namely teleportation. Furthermore, we had to cancel the dynamic walking navigation method stated in the project agreement due to not having the time to implement it as well.

### Assumptions

There were no assumptions to be held.

# Research

## Introduction

In this Chapter we discuss the problem of our project and show results of our research in the field of the application domain.

## Problem

The community provides a variety of implementation and methods for the navigation in the Virtual Reality space. Many of those couldn’t be tested and analysed scientifically. Furthermore, the already existing scientifically elaborated concepts are not necessarily suited for the new VR Hardware and the User- Space available for the VR-setup, like the HTC Vive or the Oculus Rift, and the usage in a productive application with users that have varying know-how and experience in Virtual Reality.

## Researched Navigation Methods

As described in chapter *‘2.6 Limitations and Assumptions’* we researched far more navigation methods than we could implement in the prototype. Therefore, the chapter is divided into two parts either covering the implemented methods or the various other researched navigation methods.

Each navigation methods contains the following properties:

|  |  |
| --- | --- |
| **Description** | Short description of the navigation method |
| **Physical Translocation** | Does the user need to walk in the physical space? |
| **Physical Movement** | Does the user need to do move his body in order to activate a navigation method. |
| **Parameters** | List of potentially needed parameters |
| **Problems** | List of potential problems concerning the implementation and usage of the method. |

### Implemented Navigation Methods

#### Walking in Place (WIP)

|  |  |
| --- | --- |
| **Description** | The user walks in place without changing his location in the room. |
| **Physical Translocation** | No, due to not moving in the room the physical location of the user does not change. |
| **Physical Movement** | Yes, the user needs to move his arms in a walking movement. |
| **Parameters** | * Speed * Acceleration * Deceleration * Camera Direction |
| **Problems** | * Wall Collision in the virtual reality * When does the character start to walk * Which inputs does the user have to give * Motion sickness of the user |

#### Walking by Leaning

|  |  |
| --- | --- |
| **Description** | The user leans towards the direction he wants to walk to. |
| **Physical Translocation** | No, due to not moving in the room the physical location of the user does not change. |
| **Physical Movement** | Yes, the user has to lean in order to trigger the virtual movement. |
| **Parameters** | * Location * Location (Head) * Speed * Acceleration * Deceleration * Camera Direction * Scaling |
| **Problems** | * Wall Collision * Detection of leaning degree * Scale-rate * Motion sickness |

#### Scaled Walking

|  |  |
| --- | --- |
| **Description** | The user walks inside the predefined space of the room. His physical translocation will be scaled up in the virtual reality space. |
| **Physical Translocation** | Yes, the user needs to walk in the room to activate the virtual movement |
| **Physical Movement** | Yes, the user has to lean in order to trigger the movement. |
| **Parameters** | * Location * Speed * Acceleration * Deceleration * Camera Direction * Scaling |
| **Problems** | * Wall Collision * Scale-rate * Motion sickness |

#### Pointed Teleportation

|  |  |
| --- | --- |
| **Description** | The user points towards a location he wants to teleport to. With clicking on a button he teleports to that location. |
| **Physical Translocation** | No, the user does not need to move around in the room. |
| **Physical Movement** | No, the only needed movement is to point towards a location and pressing a button. |
| **Parameters** | * Location * Camera direction * Speed of the teleport |
| **Problems** | * Camera direction after teleport (wall collision) * Camera transition |

#### Jumping

|  |  |
| --- | --- |
| **Description** | The user jumps in place. |
| **Physical Translocation** | No, the user does not need to change the location in the room in order to trigger the virtual movement. |
| **Physical Movement** | Yes, the user needs to jump in place in order to trigger the virtual movement |
| **Parameters** | * Location * Location (head) * Camera direction * Scaling |
| **Problems** | * Probably needs to be combined with other navigation methods * Physical exhaustion |

### Other Navigation Methods

#### Walking

|  |  |
| --- | --- |
| **Description** | The user walks inside a given space in the room. |
| **Physical Translocation** | Yes, the virtual location is based on the physical location in the room. |
| **Physical Movement** | Yes, the user needs to walk around in order to activate the virtual movement. |
| **Parameters** | * Location * Speed * Acceleration * Deceleration * Camera direction |
| **Problems** | * Wall collision |

#### Dynamic Walking

|  |  |
| --- | --- |
| **Description** | The user walks like in scaled Walking. The intention of the user is detected. |
| **Physical Translocation** | Yes, the virtual position is based on the user’s physical location. |
| **Physical Movement** | Yes, the user needs to walk in the physical room. |
| **Parameters** | * Location * Speed * Acceleration * Deceleration * Camera direction * Scaling |
| **Problems** | * Wall collision * Scale-rate * Motion sickness |

#### Auto Walking

|  |  |
| --- | --- |
| **Description** | The user looks down at his feet and starts to walk. |
| **Physical Translocation** | No, the user does not need to change his physical location |
| **Physical Movement** | No, the user needs only to look at his feet in order to trigger the virtual movement. |
| **Parameters** | * Speed * Acceleration * Deceleration * Scaling |
| **Problems** | * Wall collision * When does it start to walk? * When does it stop to walk? * Scale-rate * Motion sickness |

#### Walking by Button

|  |  |
| --- | --- |
| **Description** | The user presses a button on the controller to walk. |
| **Physical Translocation** | No, no physical change of the location by the user in the room needed. |
| **Physical Movement** | No, no physical movement besides pressing a button needed. |
| **Parameters** | * Speed * Acceleration * Deceleration * Scaling |
| **Problems** | * Wall collision * Scale-rate * Motion sickness |

#### Gaze-directed Teleport

|  |  |
| --- | --- |
| **Description** | The user looks towards a location he wants to teleport to. With pressing a button, he teleports to that location. |
| **Physical Translocation** | No, no physical movement required to activate the method. |
| **Physical Movement** | No, pressing a button is the only needed physical action by the user. |
| **Parameters** | * Location * Camera direction * Speed of teleport |
| **Problems** | * Camera direction after teleporting (wall collision) * Camera transition |

#### Room-to-Room-Teleportation

|  |  |
| --- | --- |
| **Description** | The user selects a room he wants to teleport to. By clicking a button, he teleports to the selected room. His location inside the room is dependent of the current location in the physical space. |
| **Physical Translocation** | No, the user does not need to walk in the physical space. |
| **Physical Movement** | No, no physical actions by the user needed. |
| **Parameters** | * Location * Camera direction * Speed of teleport |
| **Problems** | * Combining with other methods for walking in the rooms * Camera transition |

#### Zoomed Teleportation

|  |  |
| --- | --- |
| **Description** | The user looks into the direction he wants to teleport. With clicking a button, he zooms in on that location. |
| **Physical Translocation** | No, the user is not required to walk in the physical space. |
| **Physical Movement** | No, no physical actions by the user needed. |
| **Parameters** | * Location * Camera direction * Speed of zooming |
| **Problems** | * Wall collision * Camera transition |

#### Climbing

|  |  |
| --- | --- |
| **Description** | The user climbs up a wall by using his hand to pull himself up. |
| **Physical Translocation** | No, the physical location of the user does not change. |
| **Physical Movement** | Yes, the user is required to move his hand as if he is climbing up a wall. |
| **Parameters** | * Location (head) * Camera direction * Scaling |
| **Problems** | * Probably needs to be combined with another method. |

#### Flying

|  |  |
| --- | --- |
| **Description** | The user flies by using his hand / controllers like wings to navigate horizontally and vertically. |
| **Physical Translocation** | No, no translocation in the physical room required. |
| **Physical Movement** | Yes, the user uses his hand / arms like wings of a plane. |
| **Parameters** | * Location * Speed * Acceleration * Deceleration * Camera direction * Scaling |
| **Problems** | * Wall collision * Scale-rate * When does it start to fly? * Motion sickness |

#### Flying II

|  |  |
| --- | --- |
| **Description** | The user flies through the virtual world by pressing buttons |
| **Physical Translocation** | No, the user does not need to change the physical location. |
| **Physical Movement** | No, no physical movement besides pressing the buttons needed. |
| **Parameters** | * Speed * Acceleration * Deceleration * Scaling * Camera direction |
| **Problems** | * Wall collision * Motion sickness |

### Researched Parameters

The following parameters we researched are possible to be used to track and change for different Navigation methods

#### Location / Rotation (Head-Gear)

This parameter gives us access to the current location of the Head Mounted Device and can be used for scaled walking. It also gives access to the current rotation of the Head Mounted Device, which can be used for a leaning navigation method, also for determining where the User looks at.

#### Camera Direction

The camera direction can be determined from the Head Mounted Device rotation. It is used for locomotion of the user in the current gaze (camera) direction.

#### Location/ Rotation (Hand-Controller)

The Hand Controllers can especially be used to detect special navigation methods like walking in place, where the User swings his arms to move. Or even swimming motions to swim in virtual water.

#### Speed

The speed is an important factor for many movement methods, it’s critical that it does feel comfortable for the specific use case (just fast enough and just precise enough), for which it should be tested and doesn’t cause motion sickness.

#### Acceleration/ Deceleration

Acceleration and Deceleration comes to mind when working with walls that are impassable, so that the user is in a virtual environment and is blocked by walls or other obstacles when he was moving. Instead of just coming to an abrupt halt it should decelerate slowly and then stop, preventing motion sickness.

The parameter should be researched more thoroughly when the last touch for a movement Method is required.

#### Scaling

Scaling of the real world movement is possible to achieve by measuring the location change of the Head Mounted Device.

## Technical Research

The following subchapter will focus on the technical side of our research regarding the game engines and the virtual reality hardware.

### Game Engines

#### Unity 3D

Unity is a multi-platform game engine developed by Unity Technologies. It is commonly used for the development of video games for computers, consoles and mobile devices. Unity itself describes it as the world’s largest creative community and the number one game development platform[[1]](#footnote-1).

The included WYSIWYG editor makes it easy to get started and develop your first project. Another useful resource for an easy start is the rapidly growing community, a variety of tutorials and a wide range of plugins and extensions freely obtainable or purchasable in the asset store.

As for the programming language, the commonly used language is C#, but other languages like JavaScript are supported as well.

Among the normal purchasable versions, Unity offers also a free-to-use version. However, when using the free version, they automatically include a predefined Unity splash screen prior to your game. If your created game or application reaches a certain amount of revenue you are forced to get one of the paid versions. There are no royalty payments.

#### UnrealEngine4

The UnrealEngine4 is a game engine created by epic games.

One of the outstanding advantages of unreal is the blueprint system, which allows you to combine blueprints of objects and properties with functional statements in a visual way.

As for the programming language, the commonly used languages C++ and UnrealScript (a java-based object-oriented script language).

Epic Games delivers no purchasable version of the UnrealEngine4. To compensate the free usage of the engine they ask for a 5% royalty payment after reaching $3000.- of revenue per product per quarter. However, there are some exceptions for certain types of projects. « Pay no royalty for film projects, contracting and consulting projects such as architecture, simulation and visualization. »

#### Comparison & Reason of Choice

Compared to Unity 3D the UnrealEngine4 loses in the amount of supported platforms. Unity supports a wide and still growing range of platforms, while Unreal only supports the big names.

The Unity 3D Asset Store and the UnrealEngine4 Marketplace have very little in common. The Asset Store focuses on plugins, extensions and assets, while the Marketplace strongly focuses on the distribution of asset content.

Another difference between the two engines is the blueprint system of the UnrealEngine4. With this system you can create the entire project without writing code by combining blueprints with functional statements.

Due to personal reason and a greater interest we chose to work with the UnrealEngine4.

### VR Headsets

#### HTC Vive

The HTC Vive system contains the Head Mounted Device (HMD), two controllers and two base-stations.

The HMD of the HTV Vive has a visual field range of 110° (diagonally), a resolution of 2160 x 1200 overall or 1080 x 1200 for each eye and an image refresh rate of 90 Hz. The 32 built-in sensors allow for a 360° movement tracking. With the front camera it is also possible to add physical objects into the virtual world.

The measurements of the position are calculated by the two base-stations mounted to the ceiling of the room. Each base-station contains a sensor to track the position of the HMD. Furthermore the HMD has a gyroscope and an accelerometer. The two base-stations allow for a quadratic area with adjustable side length depending on the distance between the stations.

The user inputs are controlled by two hand controllers, one for each hand. The 24 sensors of the controllers allow for precise movement tracking. The multifunctional trackpad and the double-staged triggers with haptic HD-Feedback allow an entirely new virtual reality experience.

To connect the HTC Vive with a computer there are two HDMI-, two USB-, and one audio slot needed. The audio slot is needed to connect headphones to the audio slot attached to the HMD.

#### Oculus Rift

The Oculus Rift system contains the Oculus Rift, an Oculus Sensor, an Oculus Remote and an Xbox One Controller. The system can be expanded by the newly released Oculus Touch, two controllers similar to the ones the HTC Vive already has included in the base set-up.

The Oculus Rift has like the HTC Vive a visual field range of 110° (diagonally), a resolution of 2160 x 1200 overall or 1080 x 1200 per eye and an image refresh rate of 90 Hz.

The tracking of movement is measured with a gyroscope, an accelerometer and a magnetometer. The tracking of position is handled by the external Oculus Sensor.

The user inputs are handled either by a normal Xbox One Controller or the newer Oculus Touch.

#### Comparison & Reason of Choice

The two base-stations of the HTC Vive enable a wider tracking range than the Oculus Sensor The HTV Vive has a tracking range of 15 x 15 feet, while the range of the Oculus rift is limited to 11 x 5 feet.

The technical details like visual field range or resolution for both systems are more or less the same, with the only significant difference being the magnetometer in the movement tracking of the Oculus Rift.

Due to the wider tracking range and the available controllers we chose to use the HTC Vive as our virtual reality device. Another determinant factor was the better comfort of the HTC Vive Head Mounted Device.

# Implementation (Marcel)

## Introduction (Dominic)

Praktische umsetzung, protyping process

Concept and ideas

## Walking in Place

### Concept & Idea

The concept of our walking in place navigation method contains the forward / backward hand movements of a person during jogging. This gives the user the feeling of movement without physically change the location in the room. But with that comes the problem of having the feeling of moving around without moving around. To change this, we wanted to find a way to add inputs based on the leg movement when literally walking in place. However, due to the lack of leg or feet sensors this is not possible yet.

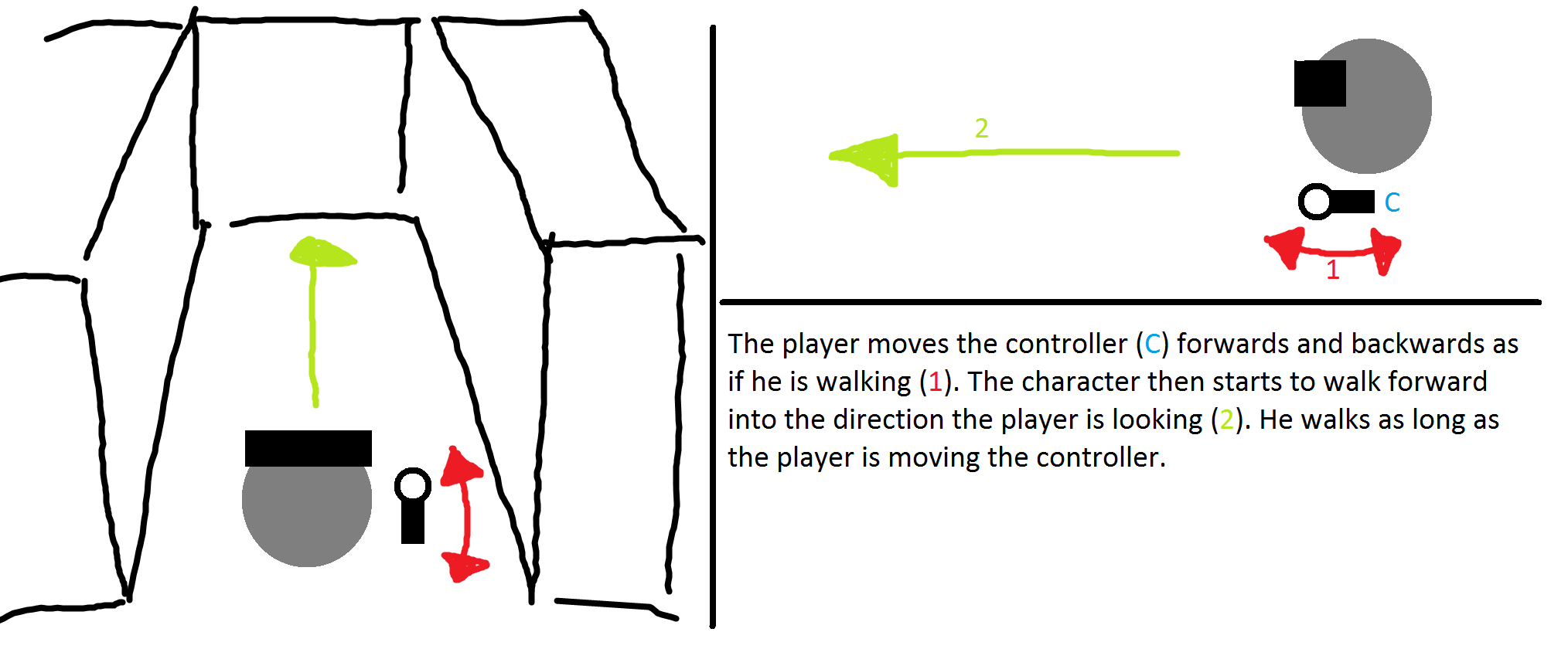


Figure 1 - Walking in place concept draft

### Parameters

The following parameters are used in this method.

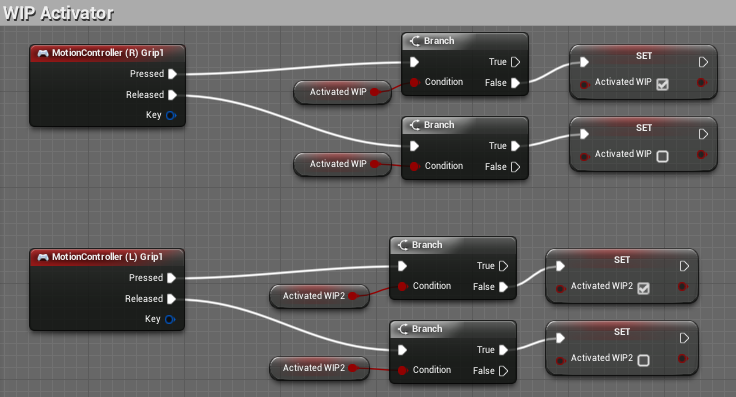
* Speed
* Camera direction
* Controller position

### Implementation

Two different implementations were prepared to answer the question in what direction the Virtual player is locomoted. One implementation used the forward direction of the controllers, to determine the locomotion direction. The second implementation used the gaze direction of the Head Mounted Display. In the following you see the blueprint implementation of the Walking in Place movement method.

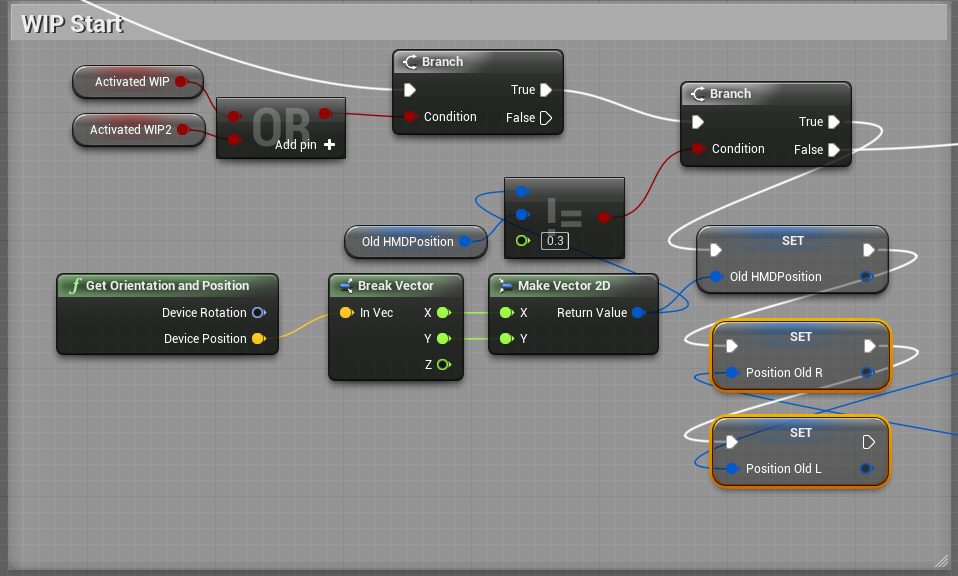
In the following we see an example implementation of the Walking in Place module that is controller oriented (the difference to HMD orientation is very small and can be found ). In the unreal project under Vive/MyViveFolder/MySimpleVive\_PawnCharacter.

Figure - Activator Walking in Place



First the walking in place needs to be activated, we used the grip button to activate it. (this is done as well for the Left MotionController.

Figure - Start Walking in Place



Then we determine if the HMD position is moved too much any movement that is done with the Left/Right Hand Controllers will be ignored. This is done to not detect any walking in place movement when the User is moving by walking in the real world.

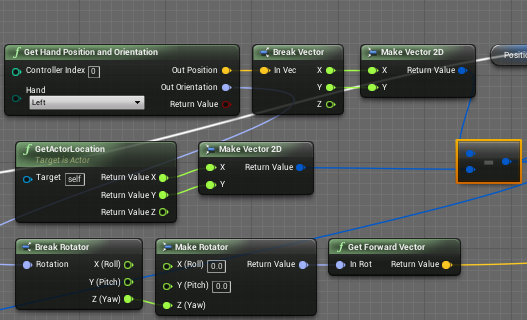


Figure 4 - Access Hand Position

We access the left Hand Position as well as the rotation and calculate the forward vector of the left Hand. We also need the Position of the Actor, that is the player’s current location, that we want to substract from the hand position, since we only want to move the position when the position of the controller is changed relative to the player’s location.

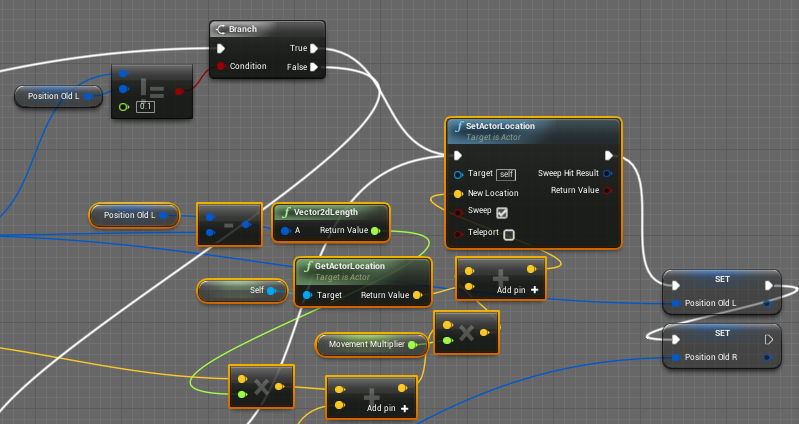


Figure 5 - Calculation of new location

As seen in the picture we compare the Left position of the controller with the old saved position of the controller if the difference between the two values is greater than 0.1, the locomotion will take place. And the Branch evaluates to true, the new location will be set and the old positions of the Left/ Right Hand Controllers will be saved.

The Marked orange part is here to calculate the new location of the player:

We see calculate the difference of the x, y value of the Left Hand controller position, then we calculate the length of the vector. This distance we multiply with the forward vector of the Left Hand Controller (we add to that the movement part calculated for the Right Hand Controller, which is similar to the above.), then we multiply with the movement multiplier, add the current location of the player to it and we have the new location that we locomoted to.

## Scaled Walking

### Concept & Idea

~~The idea of scaled walking is based on the limited physical space the user has to move, but the virtual space can be a multiple of that space. To be able to use the whole virtual space the physical movements are scaled up, so that the user can explore a multiple of the space of his physical space.~~

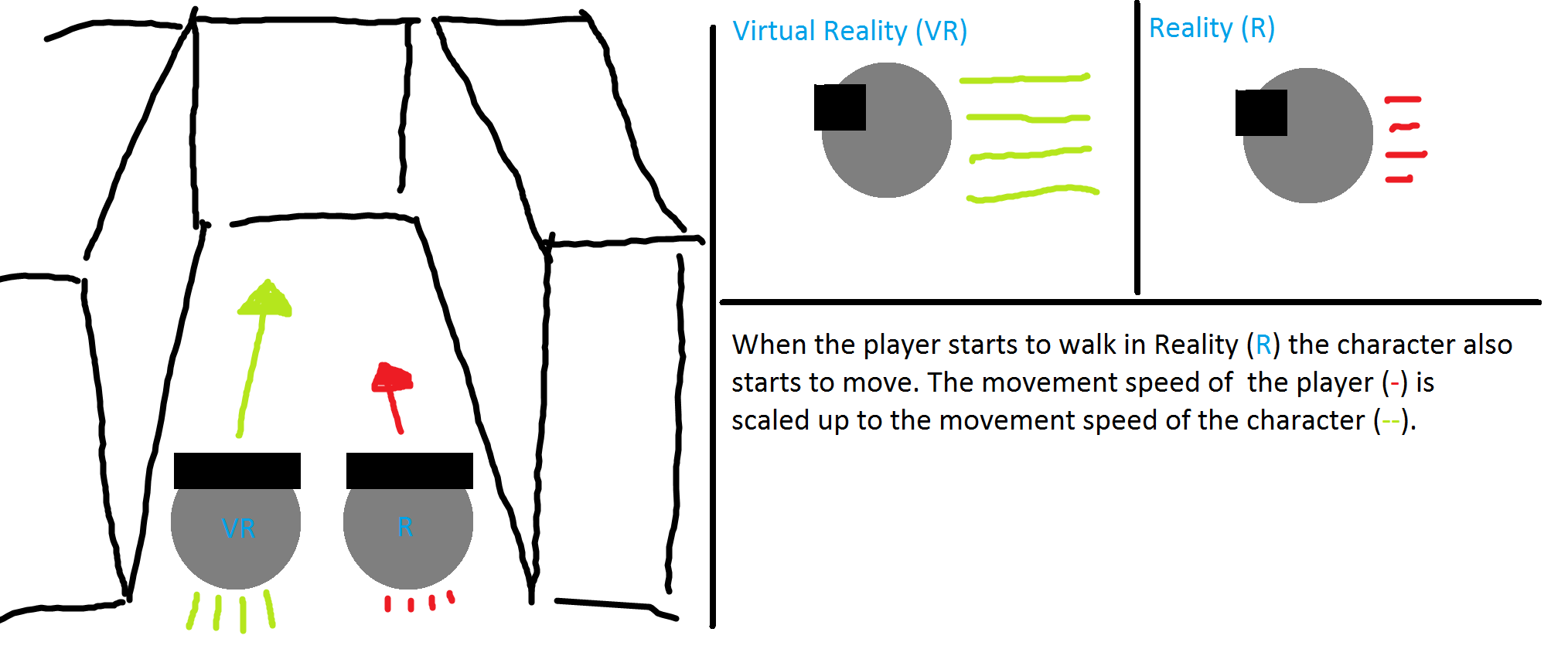


Figure 6 - Scaled walking concept draft

### Implementation

Description how it was actually implemented

Screenshot of blueprint (Different versions?)

Problems while implementing

### Parameters

Which parameters are relevant for this method

## Walking by Leaning

### Concept & Ideas

~~With walking by leaning the user leans towards a direction he wants to walk to. Once a certain threshold of the x-axis rotation is reached the virtual character begins to move into that direction. The problem with that idea is that it is more a head rotation than a full body leaning.~~

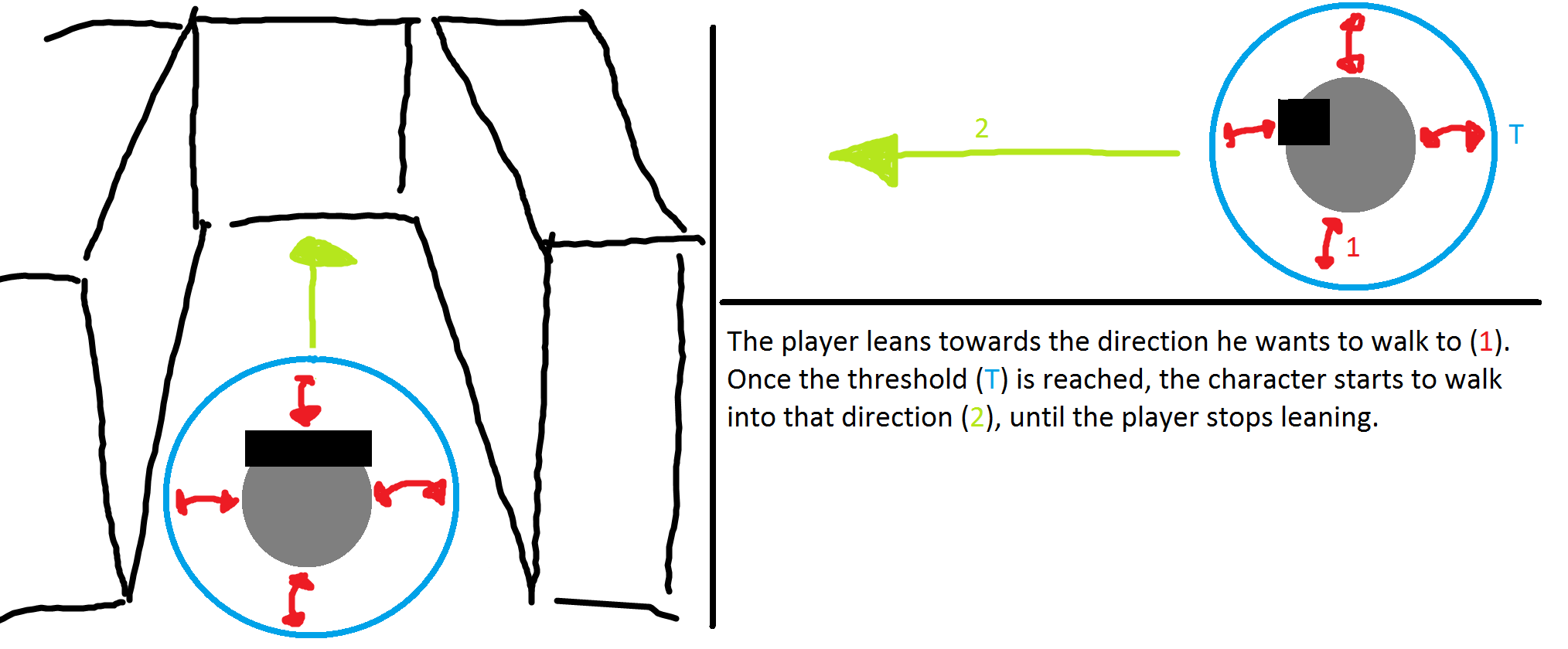


Figure 7 - Walking by leaning concept draft

### Implementation

Description how it was actually implemented

Screenshot of blueprint (Different versions?)

Problems while implementing

### Parameters

Which parameters are relevant for this method

## Jumping

### Concept & Idea

Description on how it was planned to be implemented

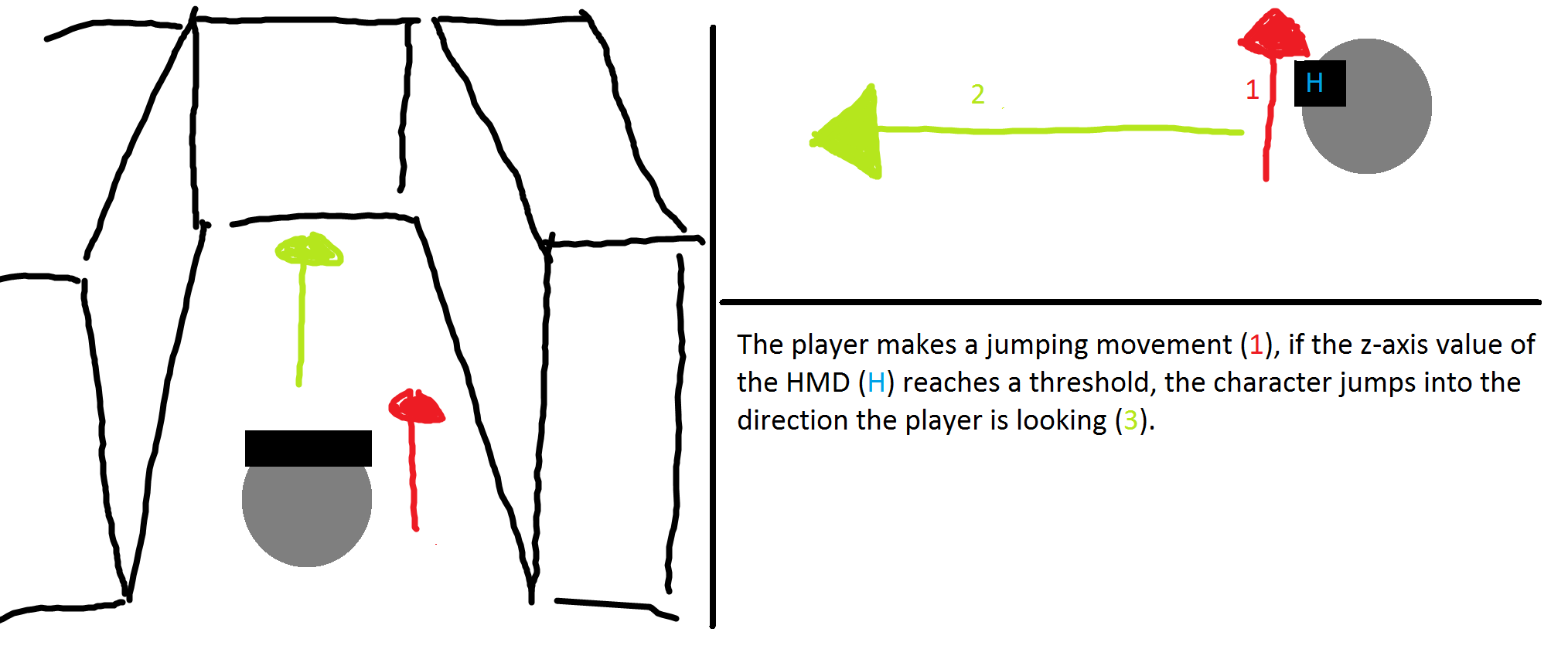


Figure 8 - Jumping concept draft

### Implementation

Description how it was actually implemented

Screenshot of blueprint (Different versions?)

Problems while implementing

### Parameters

Which parameters are relevant for this method

## Combining the navigation methods

Creation of prototype (combination of each method prototype, switch between NavMets)

# Testing (BOTH)

## Introduction (Dominic)

This chapter discusses the testing sequences. What was the scenario, what has been tested and first results will be given. A detailed discussion of the results can be found in chapter *‘6 Conclusion’*.

## Testing Szenario

The tests were hold on three different days with a total of fourteen participants. Each test was divided into five different parts (VR-Experience, Ease of Learning, Pick & Place, Jump’n’Run and Ease of Use). During each of those parts the participant was given a task to fulfilled with follow-up questions to answer. More details to each of those parts will be given in the up following chapters *‘5.3 Experience with Virtual reality’* to *‘5.7 Ease of Use’*.

The test participants are divided into two groups based on their experience with virtual reality.

## Experience with Virtual Reality

In the first test we wanted to know whether the tested person has had experience with the virtual reality prior to the test. We expected the majority to have already had first contact with the virtual reality. The results showed us that half of the test audience had had experience prior to our testing sequence.

## Ease of Learning

In this test we gave the participants time to get used to each of the four navigation methods. They had as much time at disposal as they needed to feel comfortable with the different navigation methods. We expected them to take one to two minutes to get the feeling for each method

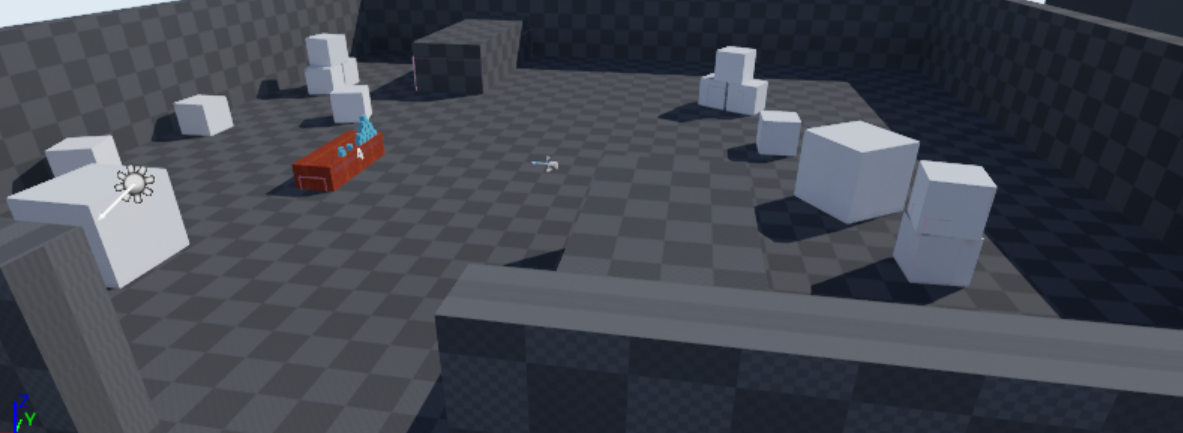
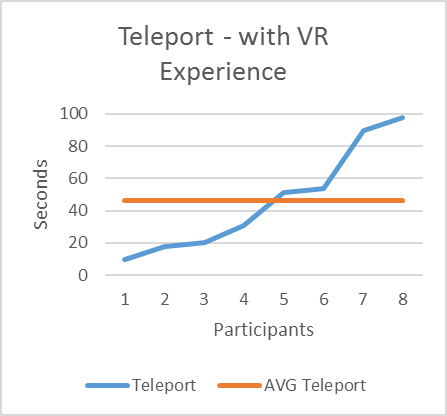


Figure 9 - Testmap Ease of Learning

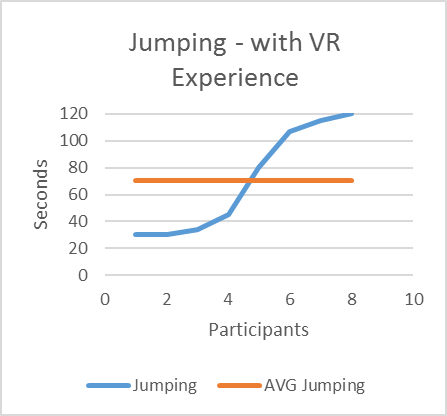
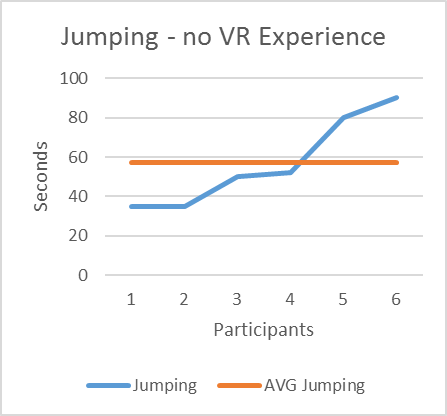
### Teleport



Chartpair 1 - EoL Teleport

As seen in the charts the average learning time for the experienced and inexperienced participants is approximately the same. However, the times of the inexperienced is closer to the average than the times of the experienced participants.

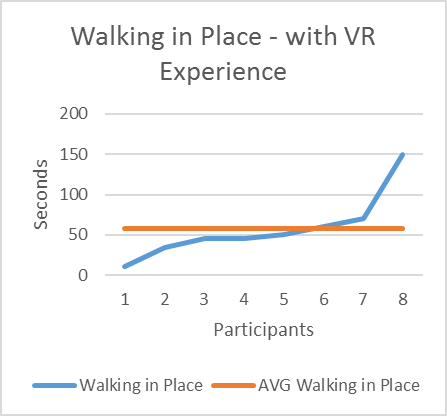
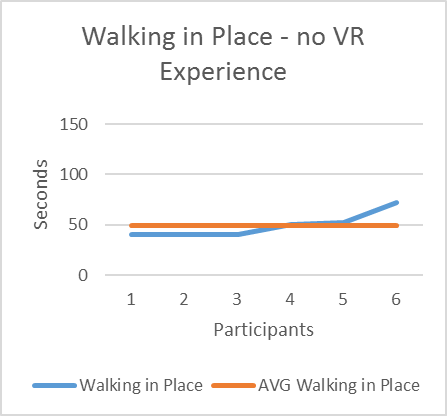
### Jumping



Chartpair 2 - EoL Jumping

Surprisingly the average learning time of the experienced participants is approximately ten seconds higher than the average of the inexperienced participants. We think this is due to them being surprised on how the method works because they’ve never seen a similar working navigation method.

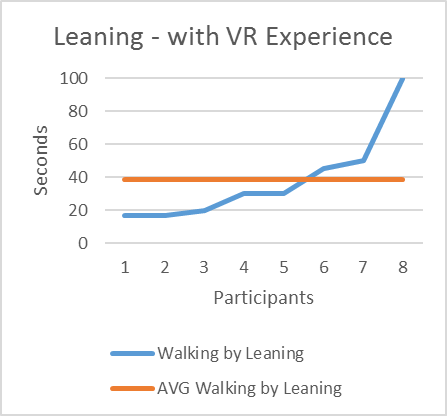
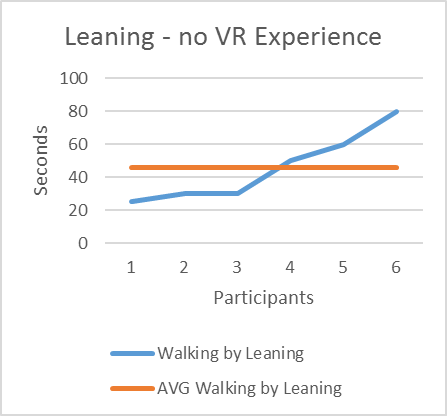
### Walking in Place



Chartpair 3 - EoL Walking in Place

Comparing both charts it is interesting to see that there is no significant difference between the two testing groups except for two participants of the experienced group spiking either with a low or a higher time.

### Walking by Leaning



Chartpair 4 - EoL Walking by Leaning

The average time of learning of the participants is between 40 to 45 seconds, which is below our expectations. However, without participant eight of the experienced group spiking off, the average of that group would be significantly below the one of the experienced group.

### Comparing all Navigation methods.

All of the different averages for each method are below our expectations. The only one being coherent with our expectations is the VR experienced jumping chart. Furthermore, the difference between the fastest and the slowest participants is higher for the experienced, while the inexperienced were closer together

## Pick & place

The test persons were asked to pick up a cube, start the timer and reach the other end of the room using the assigned navigation method. This test was done for each of the four navigation method once.

The Pick & Place task was done on four different maps. The participants were told that each map contains three different objects, but asked to not actively search for the objects. Each map has the same base lineout, the only difference being the orientation and the objects placed on the map.

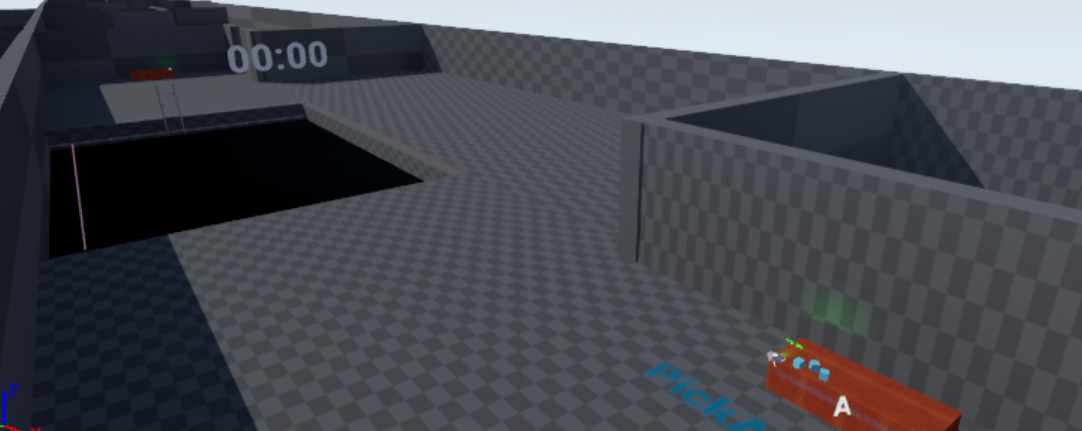
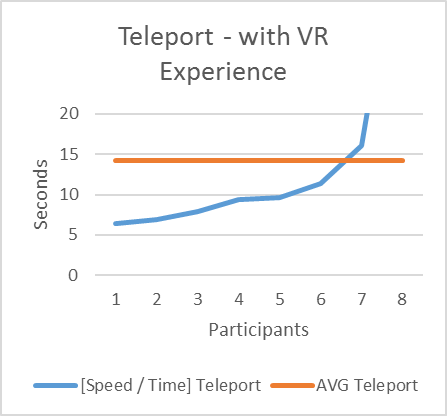
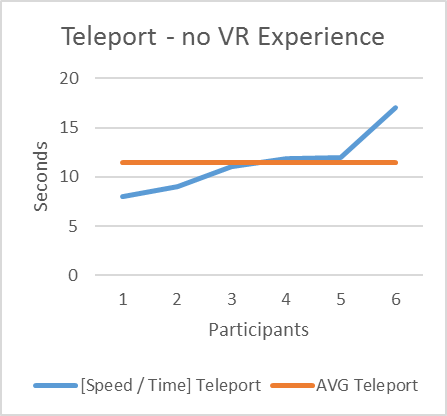


Figure 10 - Testmap Pick & Place

We estimated the following measurements

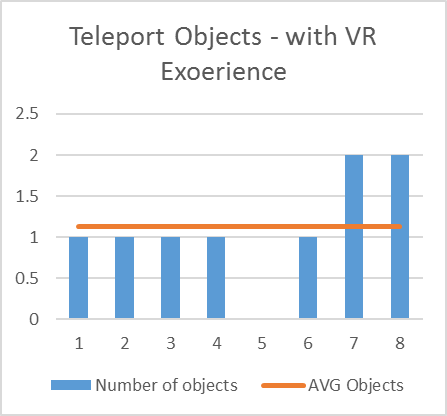
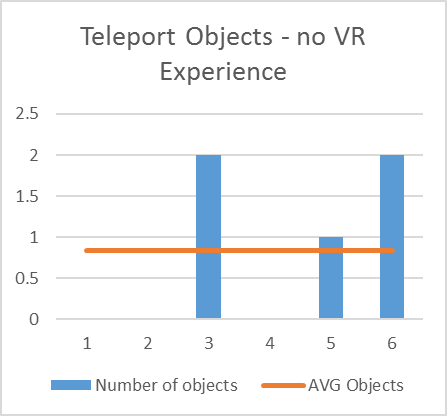
|  |  |  |
| --- | --- | --- |
| **Navigation Method** | **Time** | **Objects** |
| Teleport | 15 seconds | One object |
| Jumping | 20 seconds | One objects |
| Walking in Place | 30 seconds | Two objects |
| Walking by Leaning | 30 seconds | Two objects |

### Teleport



Chartpair 5 - P&P Teleport Time

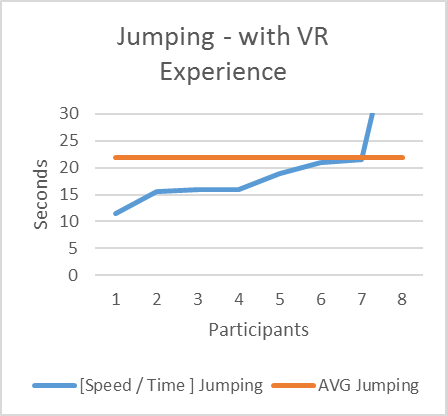
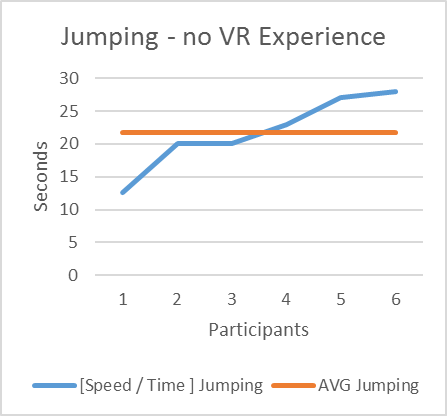
The avarages of the two groups are not comparable due to an outlier (participant 8, with VR exp). Without that outlier the average of the experienced participants would be below the average of the inexperienced. Nevertheless, the averages of both groups are slightly below our expectations of 20 seconds.



Chartpair 6 - P&P Teleport Objects

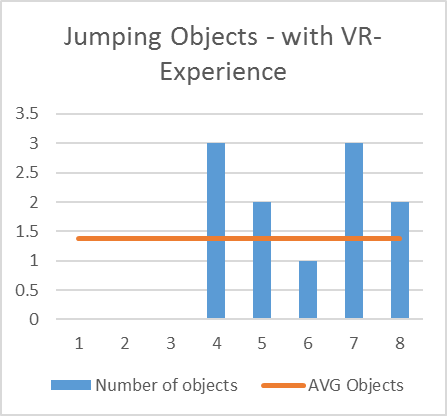
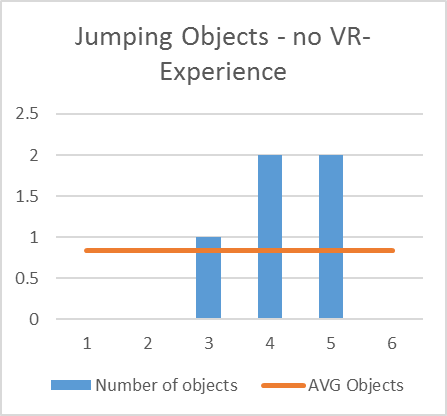
The average of both groups is slightly below / above one object, which is meeting our expectations.Furthermore, in the experienced group only one participant did not remember any object, while three of the inexperienced group failed to notice the objects.

### Jumping



Chartpair 7 - P&P Jumping Time

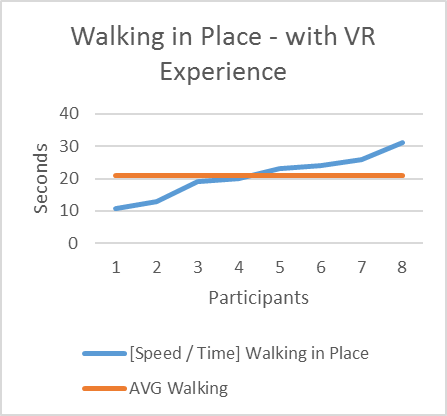
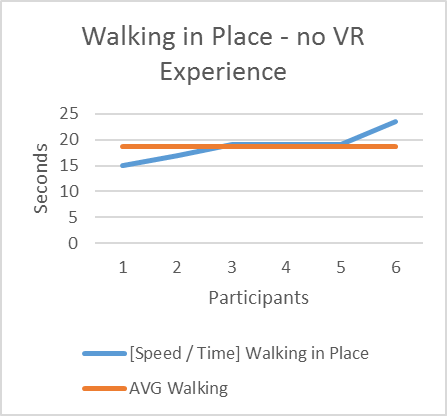
Result time



Chartpair 8 - P&P Jumping Objects

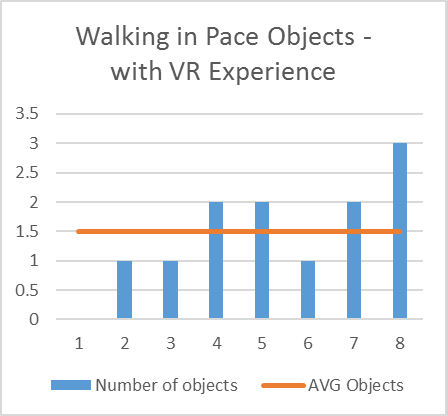
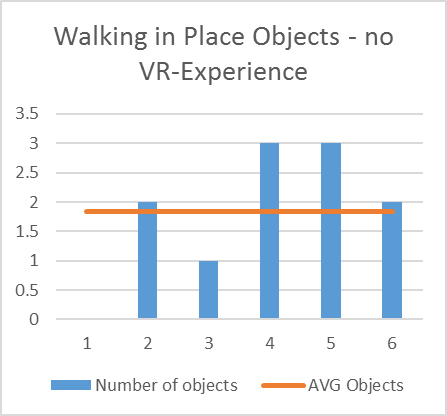
The average number of objects recognized for the inexperienced is slightly below our expectations. We think this is due to no having worked with virtual reality before and therefore not being able to adjust to the virtual environment just yet. However, there were two participants which exceeded our expectation by recognizing two objects. The Average of the experienced participants is half an object above our expectations. We were surprised that two participants were able in the short time to recognize all three objects.

### Walking in Place



Chartpair 9 - P&P Walking in Place Time

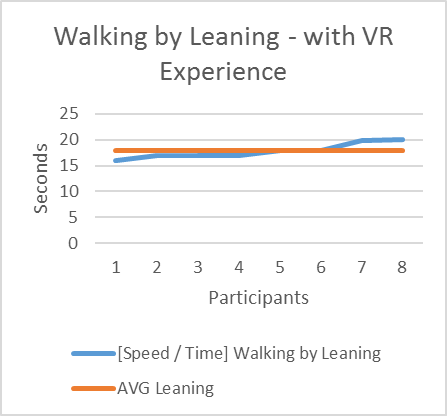
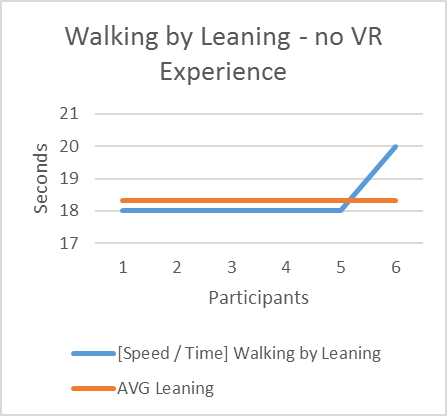
Results TIME



Chartpair 10 - P&P Walking in Place Objects

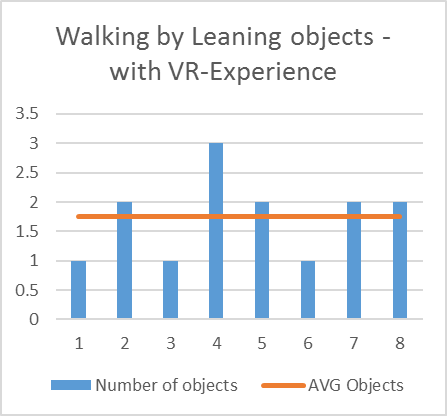
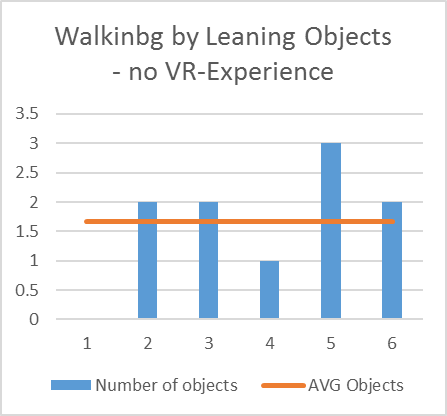
Surprisingly and in contrary to the faster navigation methods teleporting and jumping the inexperienced participants were able to recognize more objects on average than the experienced ones.

### Walking by Leaning



Chartpair 11 - P&P Walking by Leaning Time

Results TIME



Chartpair 12 - P&P Walking by Leaning Objects

Results Objects

### Overall

Results over all NavMet

## Jump’n’Run

In the third part of the test the participants were asked to jump through a pit filled with pillars to the other side. Same as in the ‘Pick & Place’ test they were asked to activate and deactivate the timer at the start and end of the course.

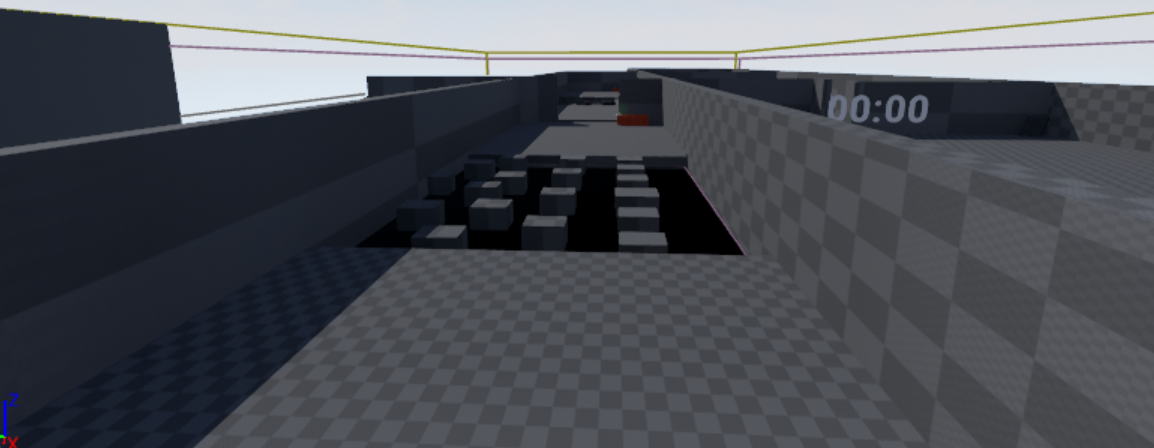
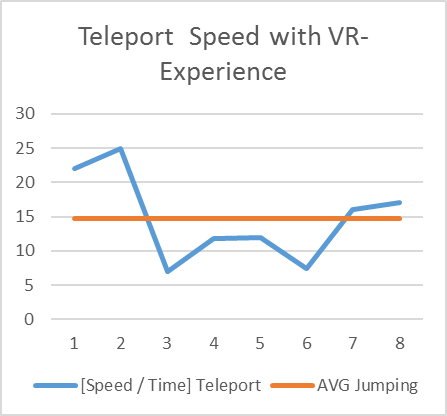
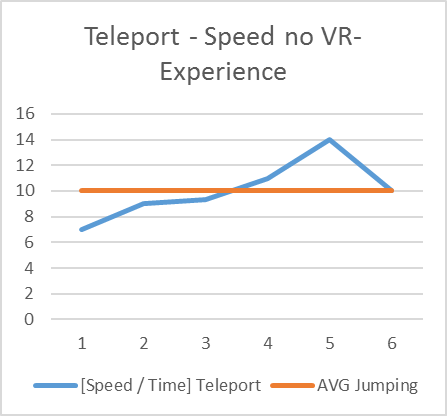


Figure 11 - Testmap Jump'n'Run

We estimated the following measurements

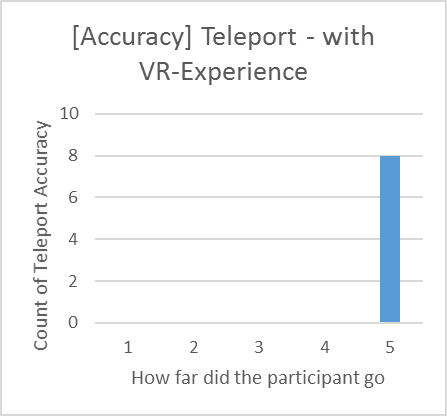
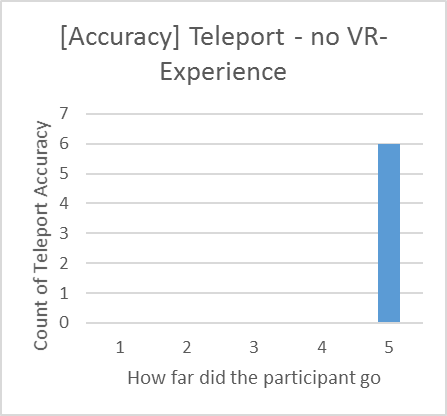
|  |  |  |  |
| --- | --- | --- | --- |
| **Navigation Method** | **Time** | **Accuracy** | **Presence** |
| Teleport | 15 seconds | 5 | 4 |
| Jumping | 60 seconds | 2 | 2-3 |

### Teleport



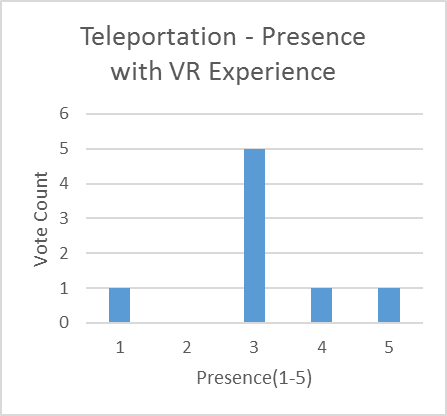
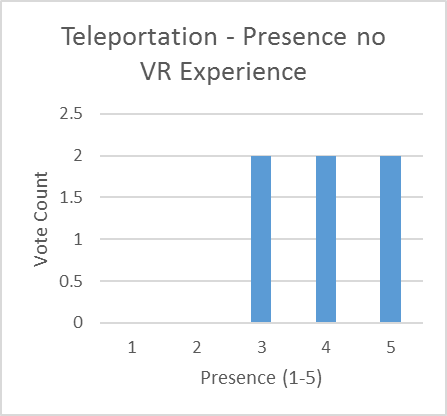
Chartpair 13 - JnR Teleport Time

Results Time



Chartpair 14 - JnR Teleport Accuracy

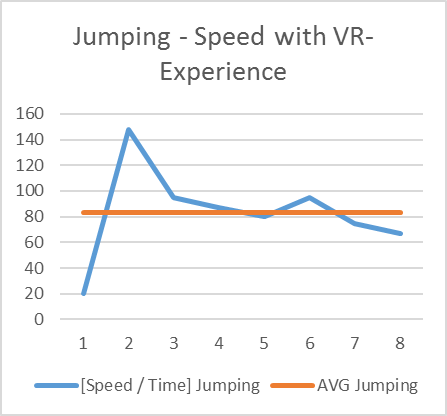
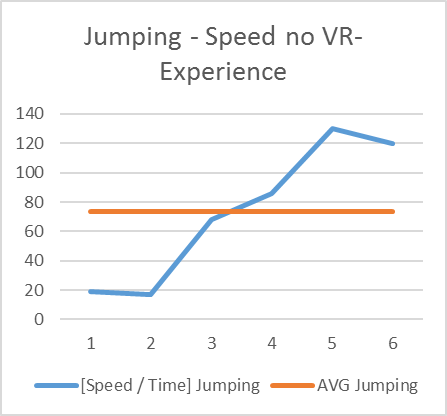
Results Accuracy



Chartpair 15 - JnR Teleport Presence

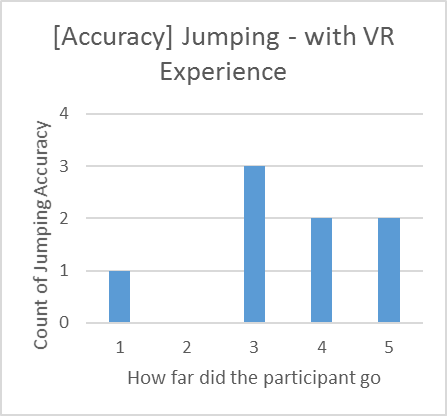
Results Presence

### Jumping



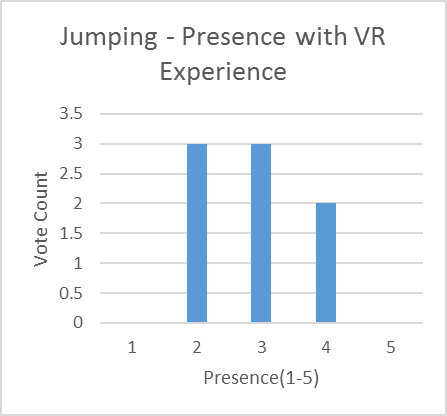
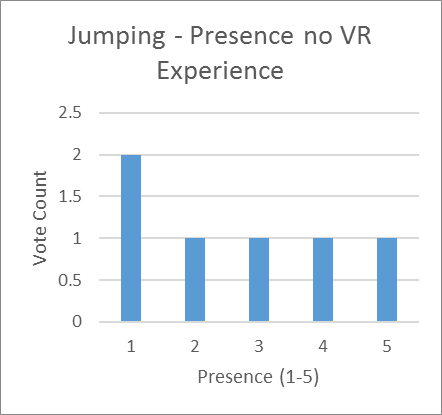
Chartpair 16 - JnR Jumping Time

Results Time



Chartpair 17 - JnR Jumping Accuracy

Results Accuracy

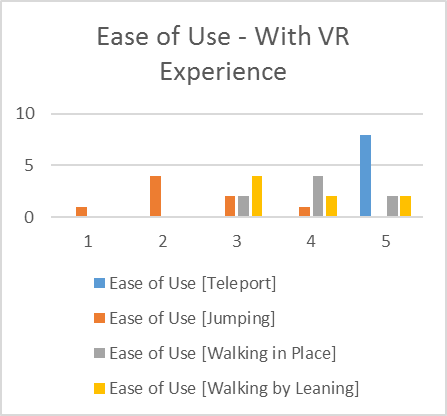
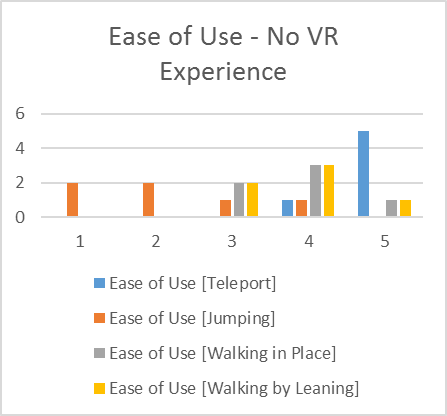


Chartpair 18 - JnR Jumping Presence

Results Presence

## Ease of Use

What tested, what expected



Chartpair 19 - EoU Navigation Methods

Results

## Problems during testing

* E.g. the ladder (or other objects) was always visible
* Button not easy usable
* Jumping is not precise enough
* Walking in Place and Walking by Leaning stops instantly when walking into an object
* By walking into a wall, the player is shifted backwards.

# Conclusion (BOTH)

## Introduction

Intro to conclusion

## Insights

We evaluated the 4 movement methods from our testing results. These parameters are specified to analyze the differences of a movement methods in virtual reality [11]. We decided to drop spatial awareness, in favor of the other parameters.

For Walking in Place and Walking by Leaning we only conducted tests for Speed, Ease of Learning, Ease of Use and Information Gathering.

The speed is difficult to measure since, there can be many appropriate velocity’s for a specific movement method, this means that instead of just a Walking in Place Method there will be a range of speed parameters that will work out in different scenarios.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Speed | Ease of Learning | Ease of Use | Information Gathering | Presence | Accuracy |
| Teleportation |  | 5 | 5 | 3 | 5 | 5 |
| Jumping |  |  |  |  | 4 |  |
| Walking in Place |  |  |  |  | - | - |
| Walking by Leaning |  |  |  |  | - | - |

Welche Erkenntnisse haben wir gemacht?

## Suggestions

There is not much experimenting with movement methods going on in the games that are publicly available. Instead you see the same kind of navigation methods in every game. The first suggestion we’ve got out of our resources that there is no fit them all navigation method, there is no navigation method that will work best with all different scenarios.

In game design you could easily introduce a navigation method first, and base the game itself on the navigation method, s.t. the game fits itself around the navigation method.

Schlussfolgerung

Konzept Suggestions which NavMet where to use

Entwicklungsprozess

# Further Steps

## Introduction

This chapter discusses various topics that could have been implemented into the project. Those topics could be implemented in a further project.

## Marketplace UE4 / Unity3D

The possibilities to create navigational assets for Unreal Engine 4 or Unity3D would help to drive the community forward, instead of just creating an asset and throwing it out there it should be researched first what is already available from the community. And then you could think about extending what is already there with what you have in mind.

## Graphical Navigation Menu / UI

A Graphical Navigation Menu should be implemented such that Users can handle the product when they come in first contact with it, so that no explanation is necessary. There is also a possibility to explain how the different methods work, e.g. with videos integrated in the UI.

## Composition of Navigation methods

The combination of different navigation methods could create composite movement methods that are put together because of their contrary strengths and weaknesses. E.g. a teleportation method that is accurate, combined together with a less accurate movement method like walking in place, which is a fast reacting movement method, in contrary to the teleportation method.

# Reflection (Both)

## Introduction

In this chapter we reflect on our project work. We will talk about what we have learned / gained, what was good or bad and our time management. Furthermore, we will reflect on the collaboration within the team and with the coaches.

## Lessons Learned

### Dominic Bär

During this project I learned how important a well-constructed time management is.. To plan and estimate how long it takes for each task to be finished and to calculate enough slack time for unexpected incidents are the key components for a successful time management.

Prior to the project I had no knowledge and experience working with the used game engine UnrealEngine4. After working with the engine for half a year now I can’t say that I mastered the creation of applications with the engine but that I learned the basics on which I can build up and further expand my knowledge. Nevertheless, I am still at the beginning on learning the potential and possibilities of the engine.

The testing was a problem we stumbled upon. At first we did not know, what and how we want to test the prototype and how much time participants need. We knew that many of our friends were interested in testing, because we were asked a lot if they could come to test and play some virtual reality games. This told us that we had interested people and that it would not be that hard to get a decent number of test participants. However, for a further project it would be necessary to know what and how you want to test while you are implementing. This would have saved us a lot of time.

Working with Marcel was nothing new for me. We worked together in the projects one and two and got to know each other. During the project he was very understanding and helped me a lot with the UnrealEngine4 problems, to understand them and to be able to fulfil my part.

### Marcel Groux

I enjoyed learning the UnrealEngine4 a lot. It was nice to get used to program with the HTC Vive and the UnrealEngine4, since there isn't that much documentation around.

There were always these strange jumps in development, as soon as we knew how to implement something it was implemented instantly in comparison to getting the know-how, which took a lot more time.

The workflow of putting the headset on and testing some navigation method, and putting it down again and putting it on again is really distracting, there are workarounds, but they are slow and annoying. I ended up strapping the HMD on and off every 10 minutes or less when I was testing methods out.

Getting used to the HTC Vive was a time consuming part, it sometimes was not working, and we needed to learn what to do when it wasn't working, after an update for the base-station I ended up reloading older firmware to the base-station, because one of them wasn't functioning properly. As we worked extensively with the HTC Vive, we shouldn't have taken for granted that it will always work.

Meanwhile the project I did had a lot of errors. It would've been better to finish the small work and then move on, but I didn't want to do the small tasks because I thought there were other priorities (challenges) that are more important or dangerous to not leave them back out unconsidered. This was an error, instead of searching for new challenges I should finish the obvious things first, this would have

given me new challenges I couldn't foresee.

It comes down to the following famous quote: "Start by doing what's necessary; then do what's possible; and suddenly you are doing the impossible." -Francis of Assisi. This is my workflow from now on.

In the project week we struggled and we were behind schedule after the first day. We should have instantly re-evaluated the situation and decide to slow the process down at least at the beginning, so we could've helped out each other. This was a big error. Deadlines that were missed during the project week, weren't right away re-evaluated. In retrospect, we should have talked right away with each other.

Later I found out that testing needs a lot of time, there is a lot to do in the preparation of the tests and also in the evaluation of the tests. I'm very grateful that we could plan and fix dates quickly, so that we could test in the same week when we were ready. In the future we should fix dates and invite people earlier, so that the timeslots we have are fully saturated.

Dominic is a forthcoming person, I enjoyed working with him a lot! I'm really grateful for all the work that he did and I'm proud to have done my part for this project.

## Time Management

Regarding the time management, we had difficulties to really estimate the needed time for the different tasks. Most of the difficulties with estimating the time for the project were based on the inexperience in the technologies. Especially hard was the calculate the time for the induction of the UnrealEngine4 and other virtual reality aspects since we did not know how effortful these tasks can get. Another difficulty was the at the beginning not defined navigation methods and the not yet clearly defined project goals. Those changed during the project when the prototype took its shape and everything was clearly defined in the project agreement. Furthermore, we had forgotten to include enough slack time in our management which lead to intensified workload during the last few weeks of the semester.

For further projects we think the time management is one of the most important tasks. Planning the project is the first step to success. Our own time management clearly needs to improve.

## Collaboration

### Team Internal Collaboration

Due to working together in the projects one and two we already knew how the other person was working and thus it was quite easy to get used to it again.

With the daily maintained Trello board we were able to get a structure in the project and an easy way to assign the various tasks to the better fitting person. In the end we divided the whole project into two parts, a theoretical and a practical to fit the personality and preferences of each of us.

### Collaboration with Coaches / Clients

The collaboration with Simon Marcin and Stefan Arisona was fine. Every week we had a meeting where we shortly discussed the progress of the project. They were motivated to give useful feedback and inputs to help us improve our work. The communication with them was very reliable.

### Collaboration with the ‘Explorative Navigation in Virtual Reality’ project team

We were able to share thoughts and ideas with Michael Läuchli and Stefan Mettler of the ‘Explorative Navigation in Virtual Reality’ project team. During the project week in the middle of the semester we shared the MediaLab and the HTC Vive to work on our projects and thus were able to collaborate by sharing insights and ideas.

# Index of Literature

L1. Internet

[1] M. Ward, R. Azuma, R. Benett, S. Gottschalk, H. Fuchs. A Demonstrated Optical Tracker With Scalable Work Area for Head- Mounted Display Systems. *Department of Computer Science Sitterson Hall University of North Carolina Chapel Hill*, NC 27599-3175

[2] S. Bendiksen, L. Jorgensen. Balancing the virtual reality experience. *Nord-Trondelag University College*

[3] R. Pausch, J. Snoddy, R. Taylor, S. Watson, E. Haseltine. Disney’s Aladdin: First Steps Toward Storytelling in Virtual Reality. *University of Virginia, Walt Disney Imagineering.* 1996 ACM-0-89791-746-4/96/008

[4] B. E. Riecke, B. Bodenheimer, T. P. McNamara, B. Williams, P. Peng, D. Feuereissen. Do We Need to Walk for Effective Virtual Reality Navigation? Physical Rotations Alone May Suffice. *Vanderbilt University, Nashville, TN, USA, Simon Fraser University, Surrey, BC, Canada, Rhodes College, Memphis, TN, USA.* Spatial Cognition 2010.

[5] T. A. Galyean. Guided Navigation of Virtual Environments. *MIT Media Lab Cambridge, MA. 02139* [*tag@media.mit.edu*](mailto:tag@media.mit.edu), 1995 Symposium on Interactive 3D Graphics

[6] J. J. LaViola Jr., D. A. Feliz, D. F. Keefe, R. C. Zeleznik. Hands-Free Multi-Scale Navigation in Virtual Environments. *Brown University Department of Computer Science*. ACM 2001 1-58113-292-1/01/01

[7] G. A. Satalich. Navigation and W**ayfin**ding in Virtual Reality: Finding the Proper Tools and Cues to Enhance Navigational Awareness. *University of Washington.*

[8] S. Frees, G. Drew Kessler. Precise and Rapid Interaction through Scaled Manipulation in Immersive. *Virtual Environments. IEEE Virtual Reality 2005*

[9] S. Regillus, E. Folmer. VR-STEP: Walking-in-Place using Inertial Sensing for Hands Free Navigation in Mobile VR Environments. *University of Nevada*. 2016 ACM. ISBN 978-1-4503-3362-7/16/05

[10] M. Slater, M. Usoh, A. Steed. Taking Steps: The Influence of a Walking Technique on Presence in Virtual Reality. *University of London*. 1995 ACM 1073-0516/95/0900-0201

[11] D. A. Bowman, D. Kollder, L. F. Hodges. Travel in Immersive Virtual Environments: An Evaluation of Viewpoint Motion Control Techniques. *Georgia Institue of Technology.* 1997 IEEE

[12] M. Usoh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater, F. P. Brooks Jr. Walking > Walking-in-Place > Flying, in Virtual Environments. ACM 1999 0-201-48560-5/99/08

L2. Existing Projects

We built up on an existing project, that already implemented movement methods, we decided to take over the teleportation method that was already implemented, and programmed our own additional navigation methods.

The project can be found under the following address:

https://bitbucket.org/mordentral/vrexppluginexample

# Index of Figures

## Figures

[Figure 1 - Walking in place concept draft 14](#_Toc472766988)

[Figure 2 - Scaled walking concept draft 15](#_Toc472766989)

[Figure 3 - Walking by leaning concept draft 16](#_Toc472766990)

[Figure 4 - Jumping concept draft 17](#_Toc472766991)

[Figure 5 - Testmap Ease of Learning 18](#_Toc472766992)

[Figure 6 - Testmap Pick & Place 21](#_Toc472766993)

[Figure 7 - Testmap Jump'n'Run 26](#_Toc472766994)

## Chartpairs

[Chartpair 1 - EoL Teleport 19](#_Toc472766995)

[Chartpair 2 - EoL Jumping 19](#_Toc472766996)

[Chartpair 3 - EoL Walking in Place 20](#_Toc472766997)

[Chartpair 4 - EoL Walking by Leaning 20](#_Toc472766998)

[Chartpair 5 - P&P Teleport Time 22](#_Toc472766999)

[Chartpair 6 - P&P Teleport Objects 22](#_Toc472767000)

[Chartpair 7 - P&P Jumping Time 23](#_Toc472767001)

[Chartpair 8 - P&P Jumping Objects 23](#_Toc472767002)

[Chartpair 9 - P&P Walking in Place Time 24](#_Toc472767003)

[Chartpair 10 - P&P Walking in Place Objects 24](#_Toc472767004)

[Chartpair 11 - P&P Walking by Leaning Time 25](#_Toc472767005)

[Chartpair 12 - P&P Walking by Leaning Objects 25](#_Toc472767006)

[Chartpair 13 - JnR Teleport Time 26](#_Toc472767007)

[Chartpair 14 - JnR Teleport Accuracy 27](#_Toc472767008)

[Chartpair 15 - JnR Teleport Presence 27](#_Toc472767009)

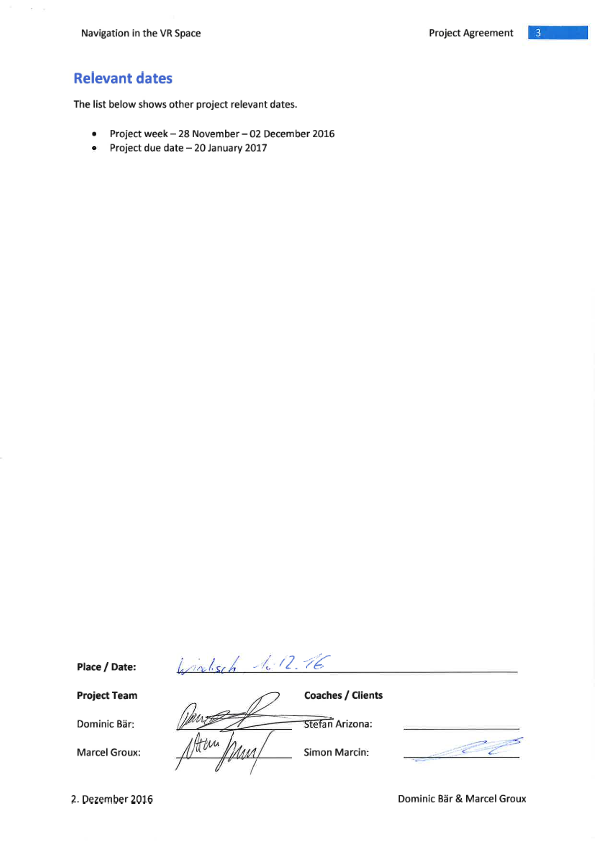
[Chartpair 16 - JnR Jumping Time 28](#_Toc472767010)

[Chartpair 17 - JnR Jumping Accuracy 28](#_Toc472767011)

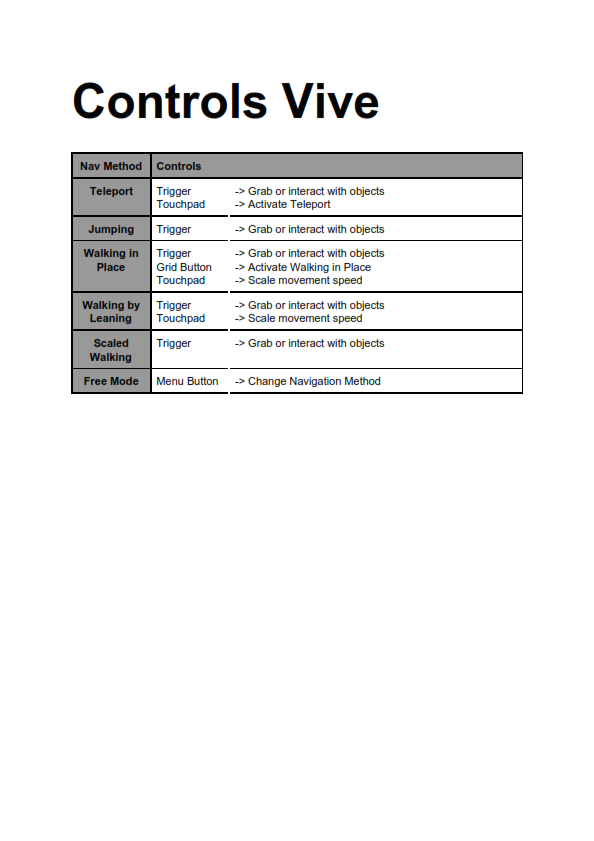
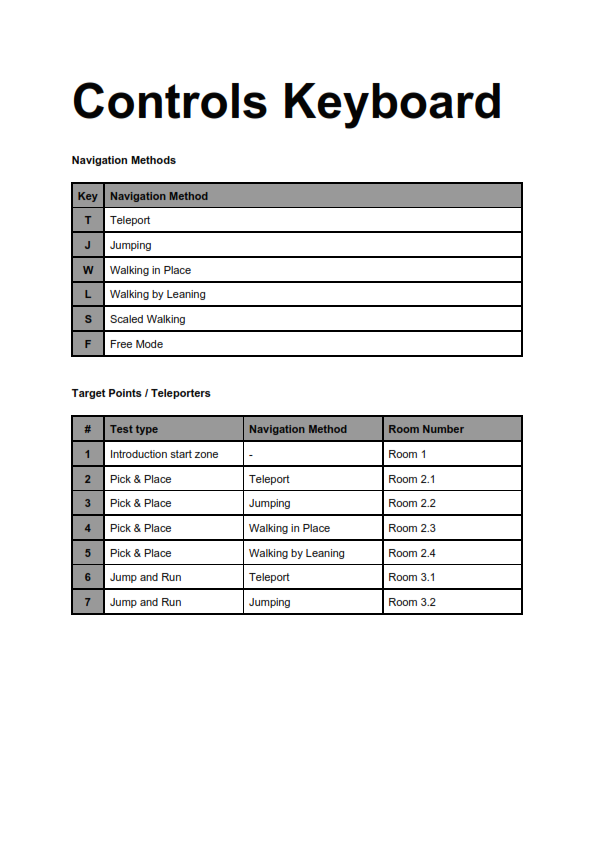
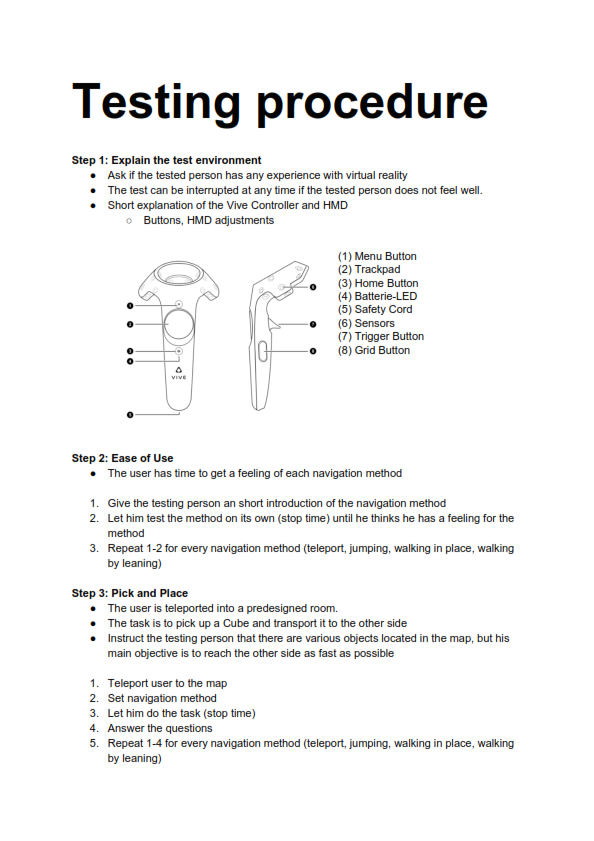
[Chartpair 18 - JnR Jumping Presence 29](#_Toc472767012)

[Chartpair 19 - EoU Navigation Methods 29](#_Toc472767013)

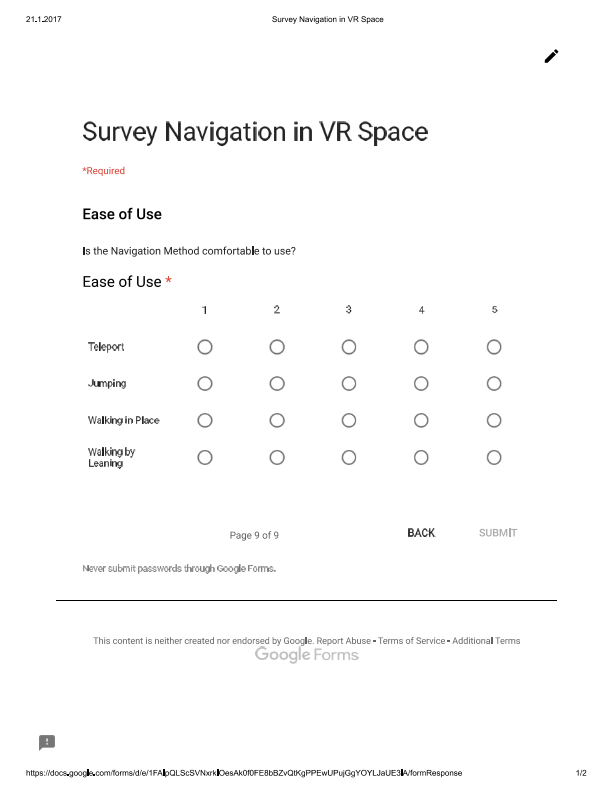
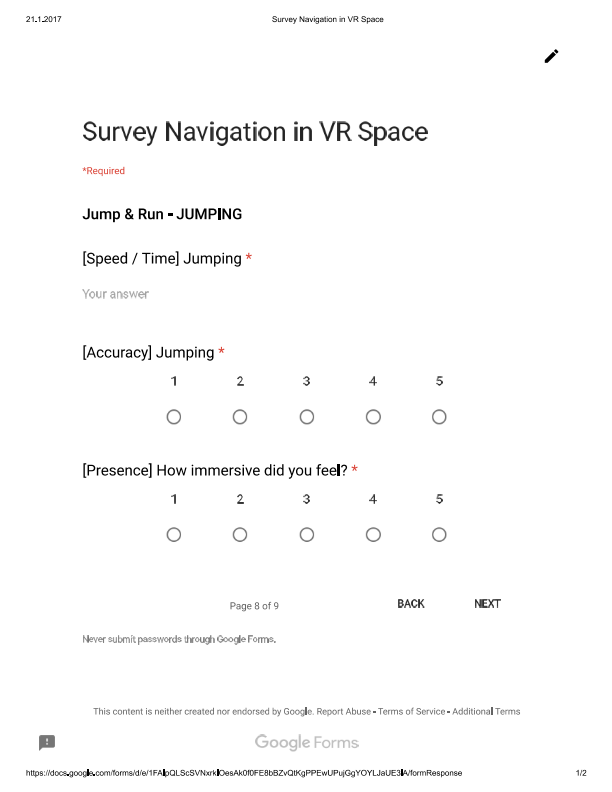
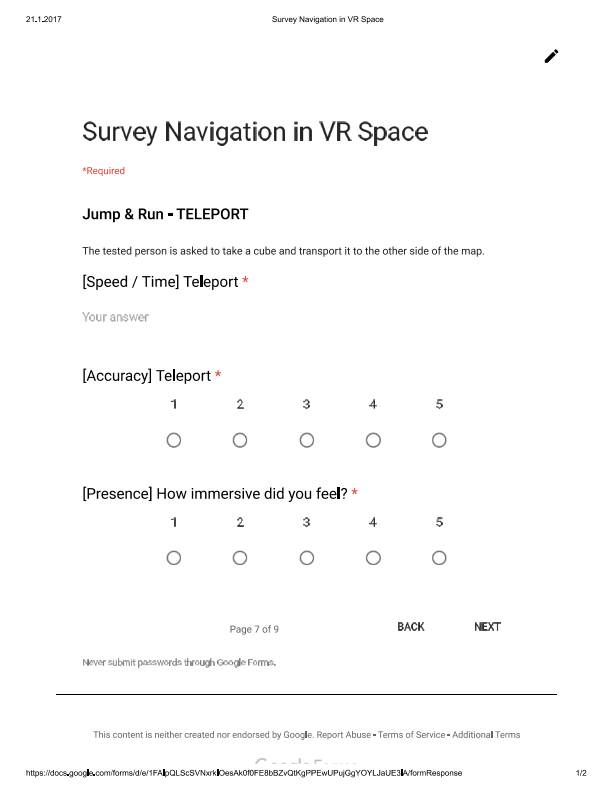
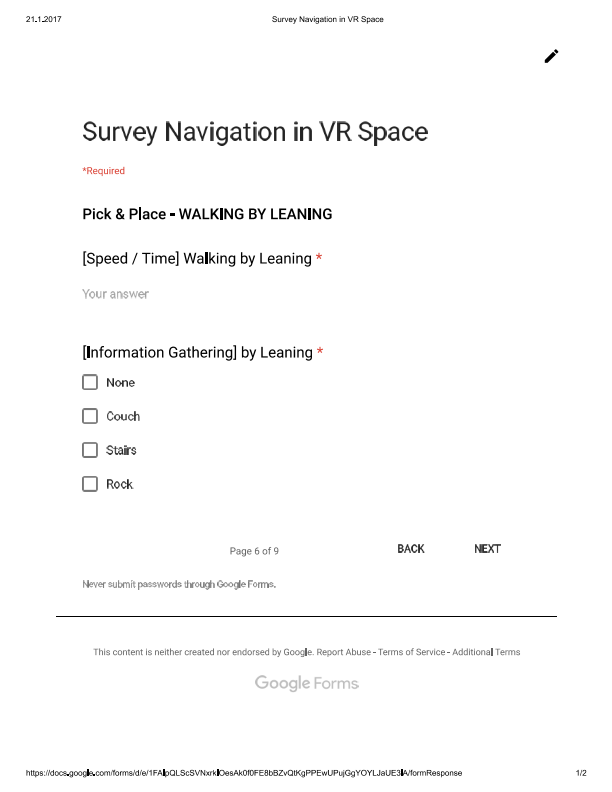
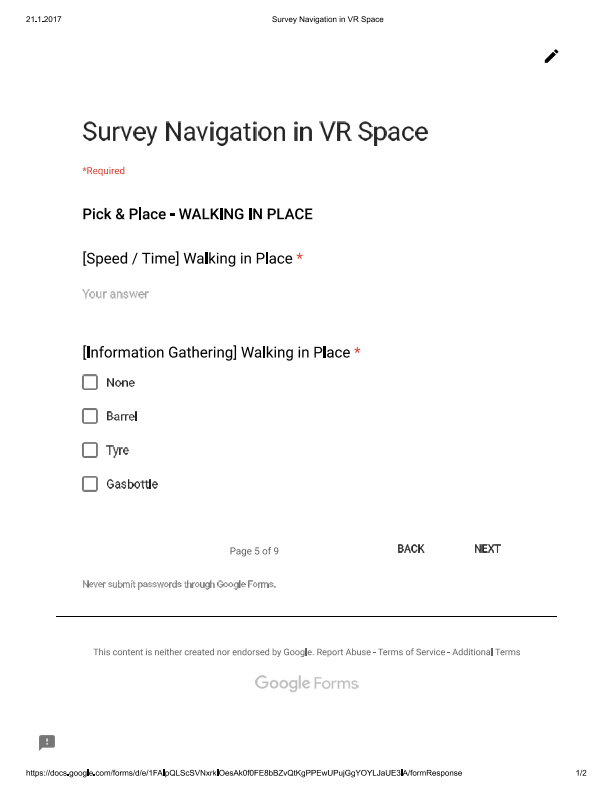
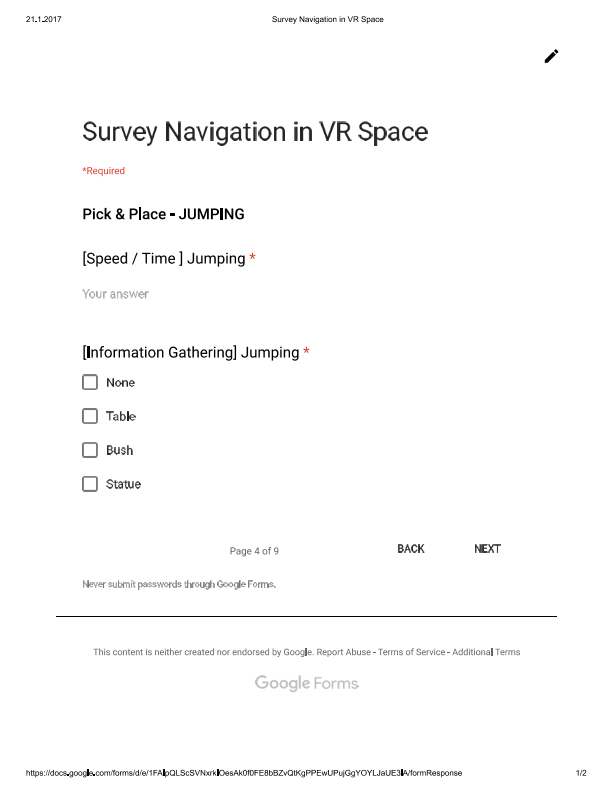
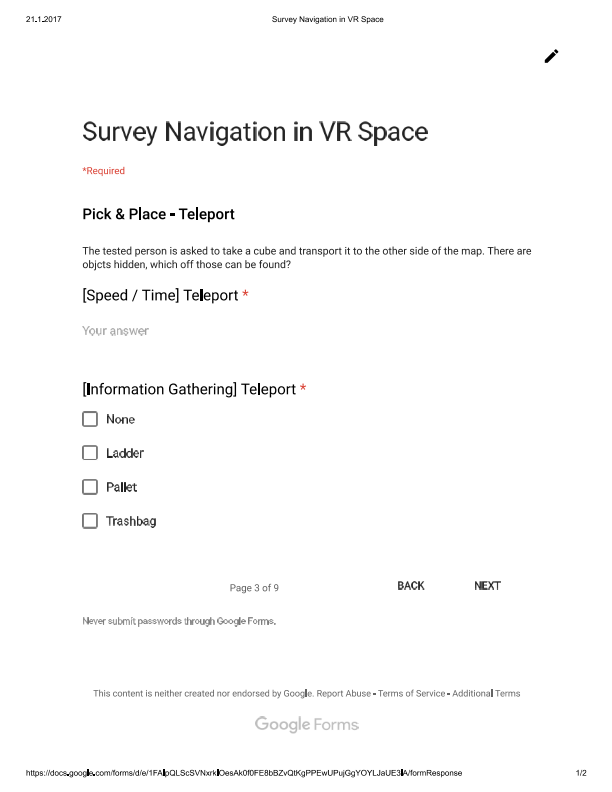
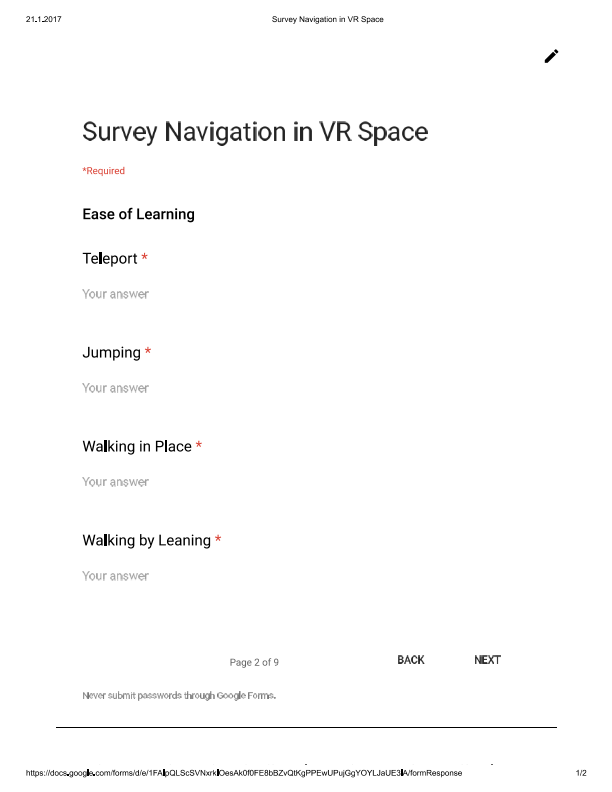
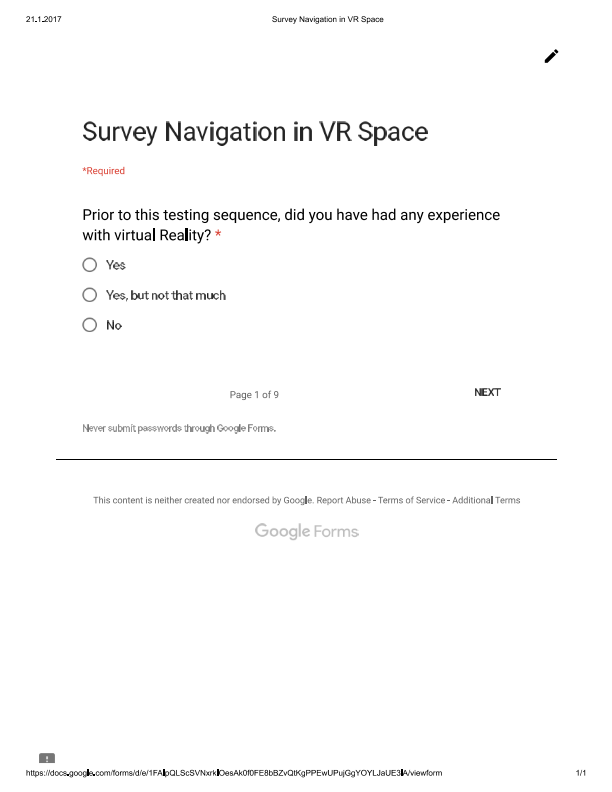
1. Attachment
   1. Project Agreement



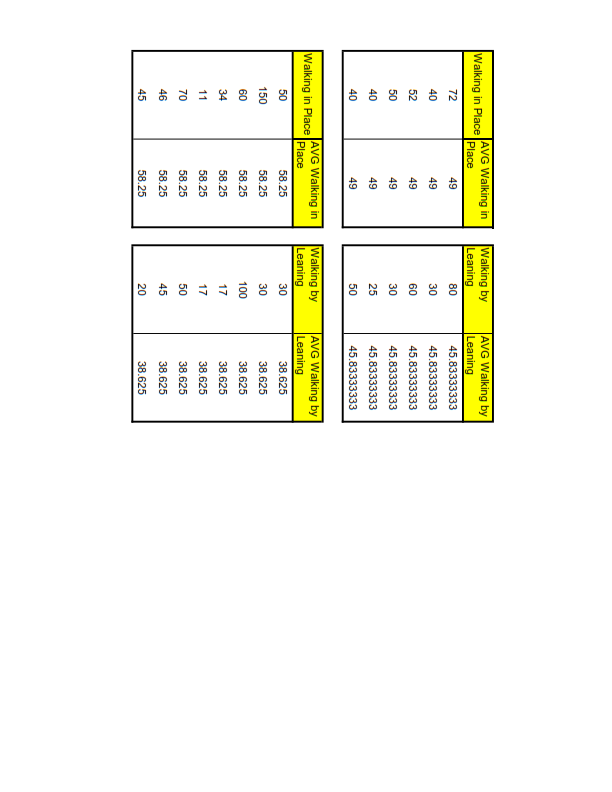
* 1. Test Procedure



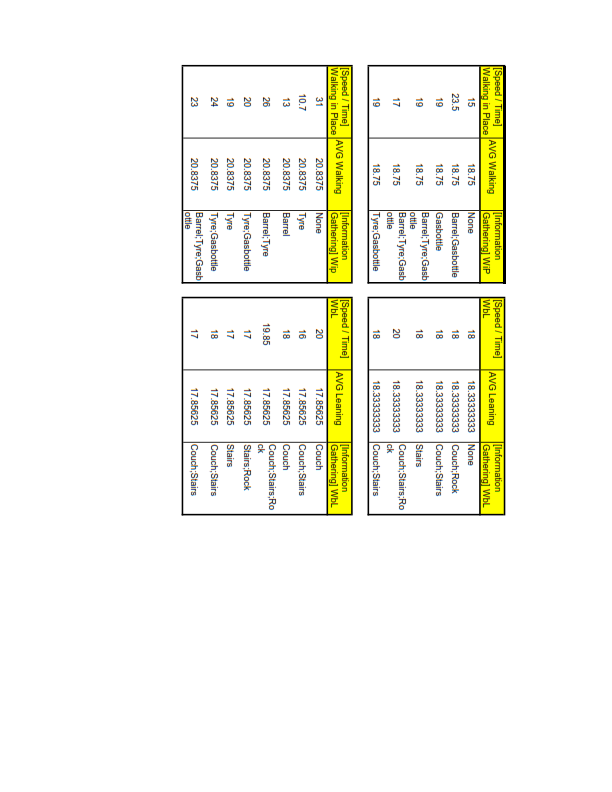
* 1. Testing Survey



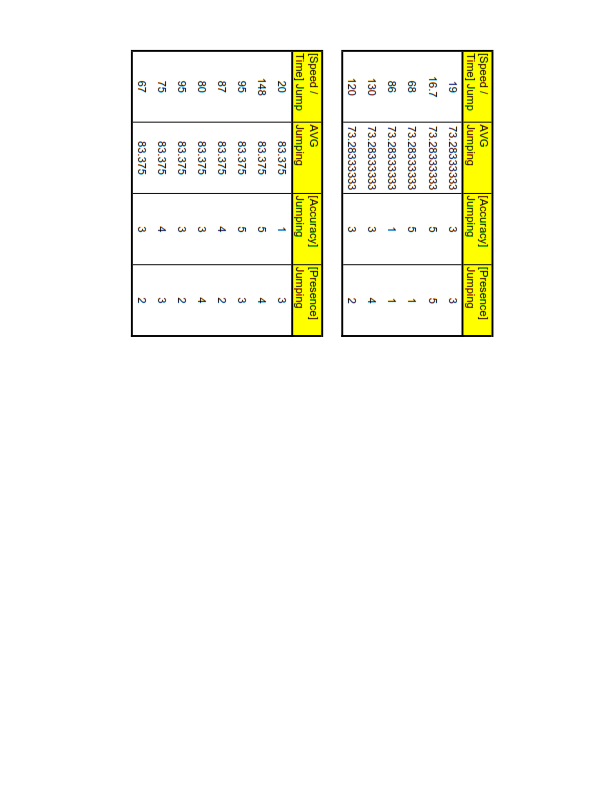
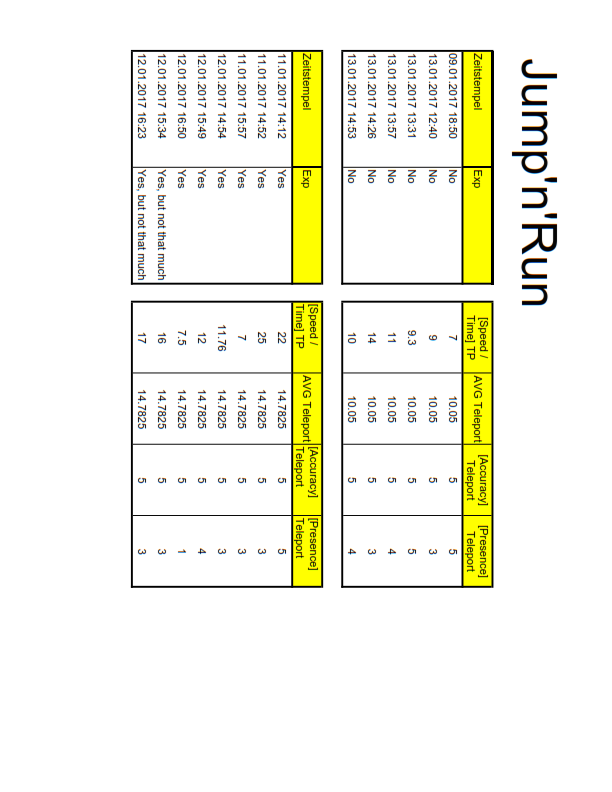
* 1. Testing Survey Results
     1. Ease of Learning



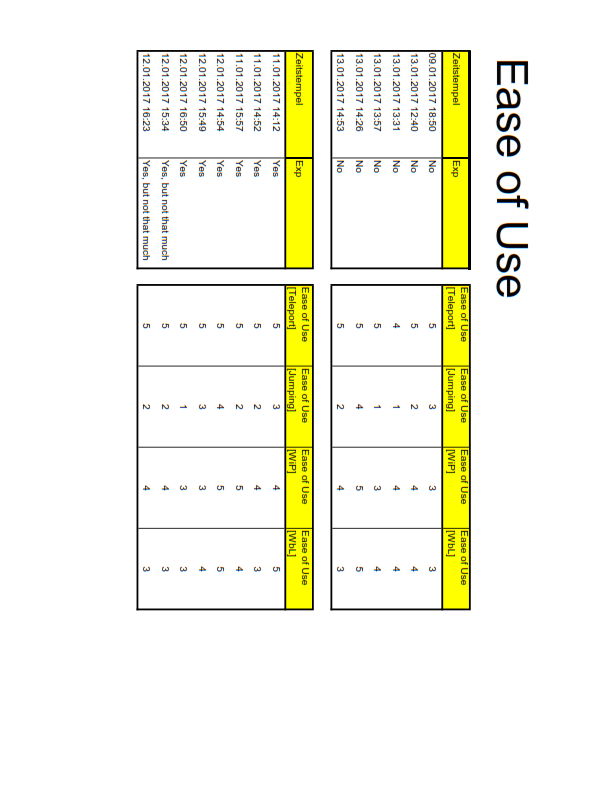
* + 1. Pick & Place



* + 1. Jump’n’Run



* + 1. Ease of Use



1. https://unity3d.com [↑](#footnote-ref-1)