

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) 5

1.1 What has been achieved? 5

1.2 Why has it been done? 5

1.3 How has it been achieved? 5

1.4 Readers Guide: How is the rest of the document constructed? 5

2 Initial Position (Dominic) 6

2.1 Introduction 6

2.2 Application domain 6

2.3 Overall scenario 6

2.3.1 Target audience 6

2.4 Project Goals 6

2.4.1 Navigation Methods 6

2.5 Project Scope 7

2.6 Limitations and Assumptions 7

2.6.1 Limitations 7

2.6.2 Assumptions 7

3 Research (Dominic) 8

3.1 Introduction 8

3.2 Problem 8

3.3 Researched Navigation Methods 8

3.3.1 Implemented Navigation Methods 9

3.3.2 Other Navigation Methods 11

3.3.3 Researched Parameters (MARCEL) 14

3.4 Technical Research 14

3.4.1 Game Engines 15

3.4.2 VR Headsets 16

4 Implementation (Marcel) 17

4.1 Introduction (Dominic) 17

4.2 Walking in Place 17

4.2.1 Concept & Idea 17

4.2.2 Implementation 17

4.2.3 Parameters 17

4.3 Scaled Walking 18

4.3.1 Concept & Idea 18

4.3.2 Implementation 18

4.3.3 Parameters 18

4.4 Walking by Leaning 18

4.4.1 Concept & Ideas 18

4.4.2 Implementation 19

4.4.3 Parameters 19

4.5 Jumping 19

4.5.1 Concept & Idea 19

4.5.2 Implementation 20

4.5.3 Parameters 20

4.6 Combining the navigation methods 20

5 Testing (BOTH) 21

5.1 Introduction (Dominic) 21

5.2 Test 21

5.3 Expected results 21

5.4 Test Results 21

5.5 Problems during testing 21

6 Conclusion (BOTH) 22

6.1 Introduction 22

6.2 Insights 22

6.3 Suggestions 22

7 Further Steps (Marcel) 23

7.1 Introduction (Dominic) 23

7.2 Marketplace UE4 / Unity3D 23

7.3 Graphical Navigation Menu / UI 23

7.4 Composition of Navigation methods 23

8 Reflection (Both) 24

8.1 Introduction 24

8.2 Lessons Learned 24

8.2.1 Dominic Bär 24

8.2.2 Marcel Groux 24

8.3 Time Management 24

8.4 Collaboration 24

8.4.1 Team Internal Collaboration 24

8.4.2 Collaboration with Coaches / Clients 25

9 Index of Literature (Both) 26

L1. Literature 26

L2. Internet 26

L3. Existing Projects (Marcel) 26

10 Index of Figures 27

A. Attachment (Dominic) 28

A1. Project Agreement 28

A2. Test Survey 28

A3. Attachment 3 28

A4. Clarification of Honest 28

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 covering the two main groups of navigation methods, teleporting and walking.

To further use this prototype in upcoming projects a concept containing suggestions for using the different navigation methods.

## Why has it been done?

The prototype was created to analyze 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 research many different navigation methods and their used parameters. Based on those we choose a number of navigation methods we wanted to implement and created a concept and idea how we imagined them to be implemented. The second part covers the implementation and various self-tests on a weekly basis 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 separate parts. The first one contains the theoretical aspects covering the problem and the research. The second part addresses the practical aspects of implementation and testing.

1. Initial Position (Dominic)

## 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

**Anwendungsdomäne**

## 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 and / 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 3.3 Research 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 choose the navigation methods that are commonly used and those that interested us personally the most.

### Assumptions

Welche Annahmen mussten wir treffen?

1. Research (Dominic)

## 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 however couldn’t be tested and analyzed 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 vertivally. |
| 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 (MARCEL)

These parameters that we researched are possible to track and change for different Navigation Methods.

* Location / Rotation (Head-Gear)

Access to the current location of the Head Mounted Device can be used for scaled walking.

Access to the current rotation of the Head Mounted Device 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 inpassable, s.t. the user is in an 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-plattorm 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 usefull 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 Head Mounted Device (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 taken 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. The position of the HMD is measured with 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 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

Description of Oculus Rift

#### Comparison

# Implementation (Marcel)

## Introduction (Dominic)

Praktische umsetzung, protyping process

Concept and ideas

## Walking in Place

### Concept & Idea

User swings his arms, like he would do when walking, the controllers are than tracked for the distance which is covered by them and the Virtual player is then locomoted.



Figure 1 - Walking in place concept draft

### Parameters

Speed, Camera Direction, Controller Position

### Implementation

2 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.

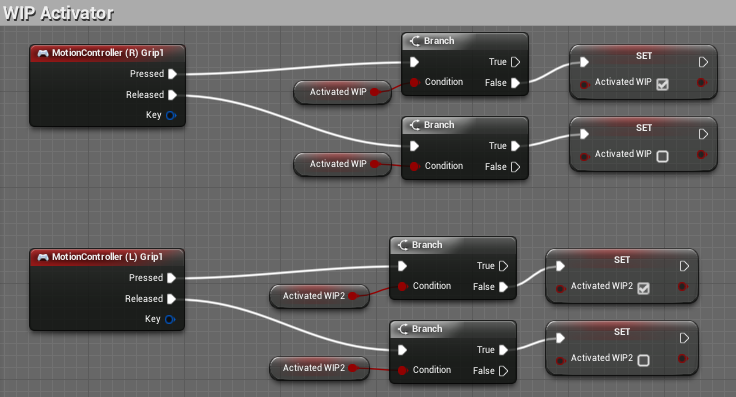
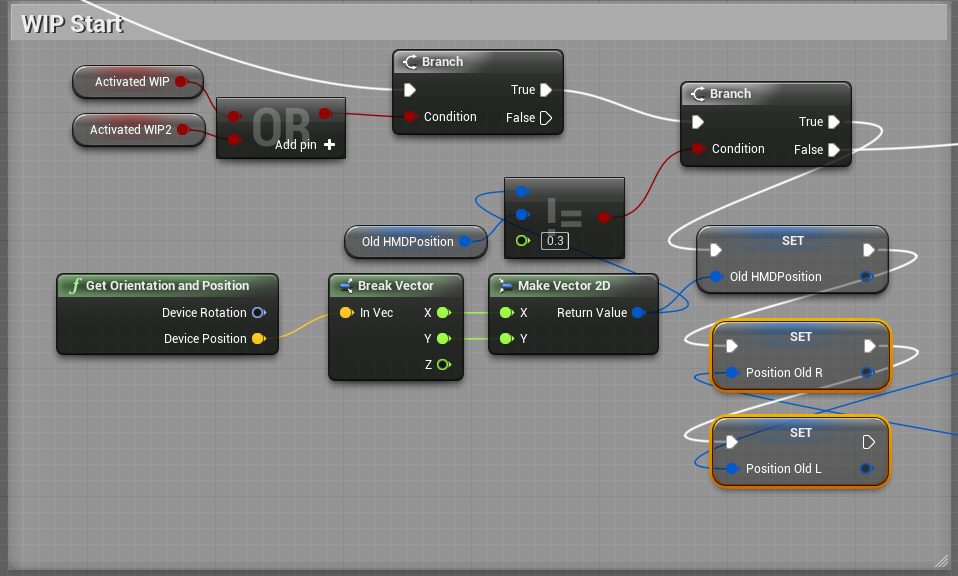
First the walking in place needs to be activated, we used the grip button to activate it. (this is done as well for the Left MotionController.

Figure 2 - Activator of 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 3 - Start of Walking in Place

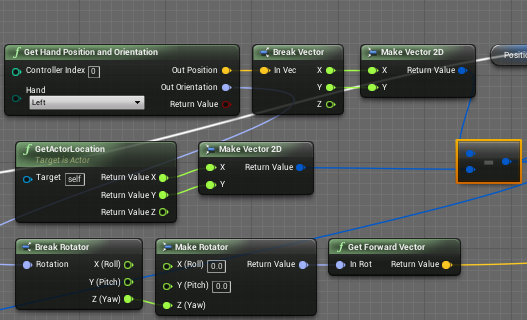


Figure 4 - Accessing Hand Position and Actor Position

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 players 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 players location.

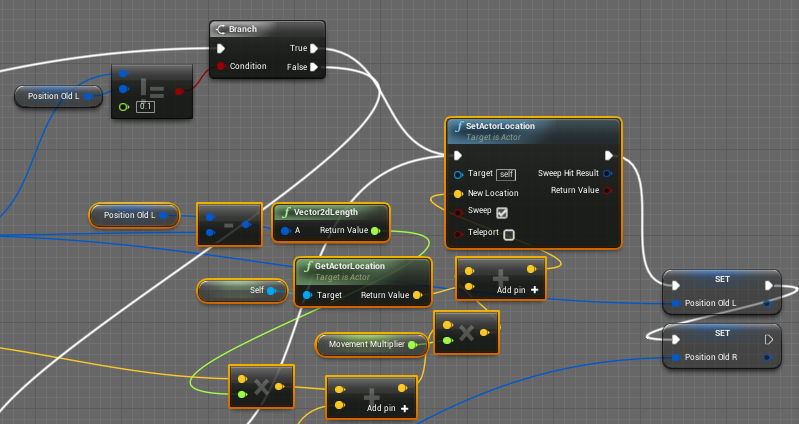


Figure 5 - Locomotion of Walking in Place

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

Real world movements are scaled up to the virtual movements. So we can travel by foot in the play area that we have in large rooms.



Figure 6 - Scaled walking concept draft

### Parameters

Speed (movement multiplier), Head Mounted Device Location

### Implementation

The difference of the Head Mounted Device location is measured and is then translated into locomotion.

The main question here comes down to when to enable scaling, other questions came up during implementation, these are if it would make sense to implement a gaze direction scaling.

And these values or more critical than in the other movement methods, since the behaviour is coupled with real walking. And this results in a distorted image if wrong decisions are made, the resulting is motion sickness. It’s unclear how to prevent motion sickness with this method.

The Scaling method was left out of the tests, since we have not found a reliable not motion sickness inducing way to move through the virtual world.

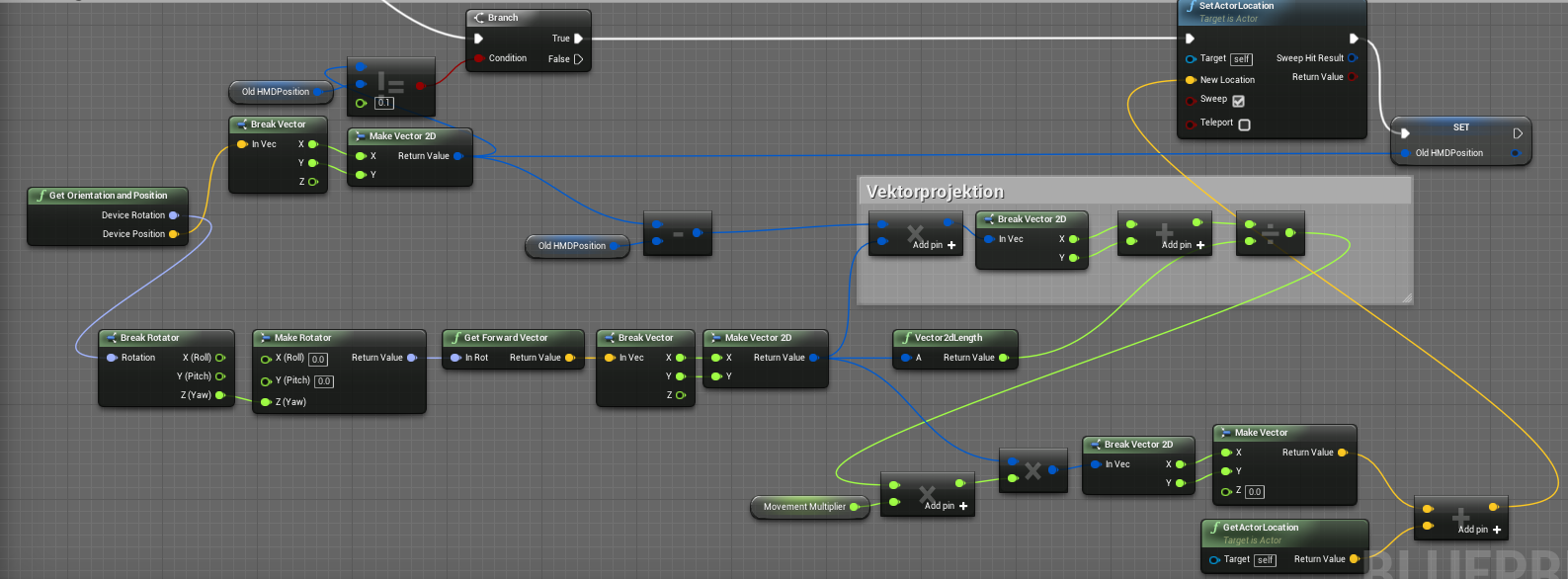


Figure 7 - Scaling with vector projection

We activate this scaling method, when the Head Mounted Device gets moved over a threshold. This method allows only movement in gaze direction, which could be interesting for specific use cases. With a large multiplier, large distances can be travelled fast, but only in one direction, this means, you need to turn around or implement a method with which you can turn around in the virtual world, so you can turn around in the real world, and walk further.

## Walking by Leaning

### Concept & Ideas

When the user leans forward, he will glide forward in the virtual world. This could be very well implemented into a game concept, s.t. the movement is a special movement from a character you control (for example the Reaper from Overwatch).



Figure 8 - Walking by leaning concept draft

### Parameters

Speed, Head Mounted Device Rotation and Location, Controller Location.

### Implementation

There are multiple versions that we implemented, we experimented first with a version that used only the rotation of the Head Mounted Device to check if the user was leaning.

The most important question here is, when does the player start to move?

In the first version we tried out different activation values for the angle of the Head Mounted Display. After that we discovered that, since the player is leaning extensively, he can’t really see where he’s going, if we set the activation value to near to the normal looking angle, the player would move unexpectedly.

Therefore we implemented a version that we called “Batman”, it locomotes the User when the Controllers are behind the Head Mounted Device forward Vector (that is the gaze direction of the user).

In the following section we are going to see the implementation of the Walking in Place Method called “Batman”.

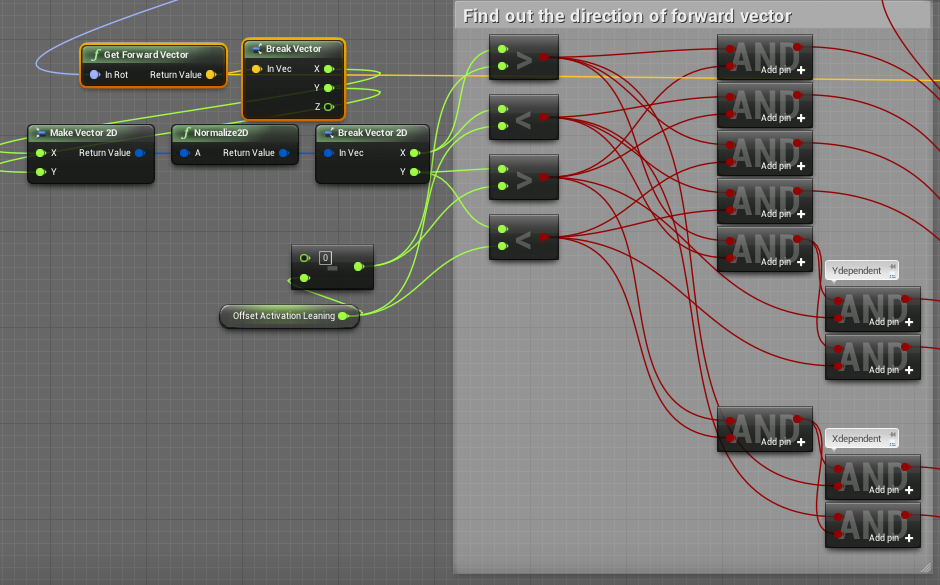


Figure 9 - Finding the forward vector of the HMD

We find out what has to be true, when the vector of the HMD is pointing in a certain direction, there are only 4 variants, however there are 4 more variants for when it’s only X dependent or only Y dependent, then the controllers position has to be either only greater or smaller for the X position or only greater or smaller for the Y position.

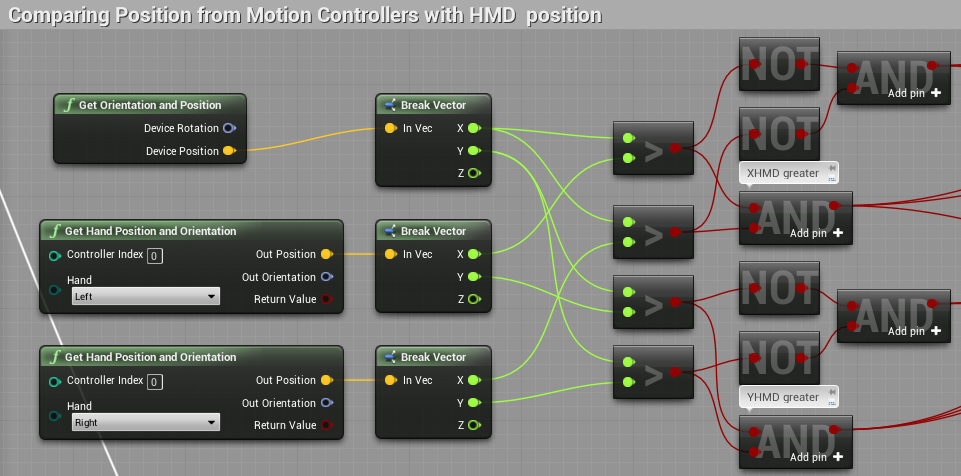


Figure 10 - Comparing position Controllers and HMD

Then we compare if the actual position of the HMD is either greater or smaller to the Y and X Position of the controllers.

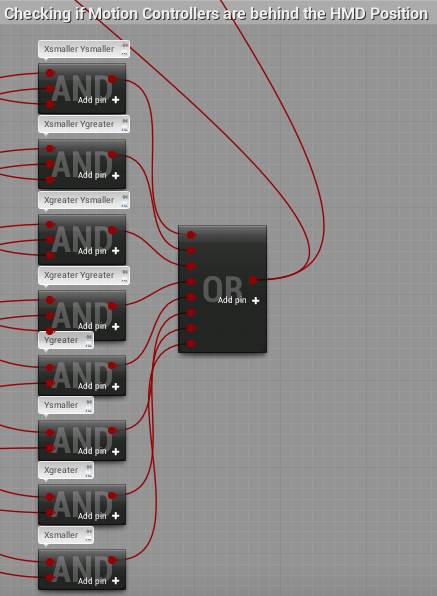


Figure 11 - Are Motion Controllers behind the HMD?

In the third step we put everything together and we activate the Walking by Leaning locomotion if any of the AND’s evaluate to true.

## Jumping

### Concept & Idea

When the User jumps, the position of the user changes a certain distance forward. There can be many use cases for this, a jump and run could be made like this.



Figure 12 - Jumping concept draft

### Parameters

Head Mounted Device Location and Rotation(Camera Direction), Time

### Implementation

After an initial implementation we decided to use an jump indicator for the User. It is spawned in the direction of gaze of the player.

I came to an implementation problem that I couldn’t get around, I tried to fix the position of the indicator to the camera of the player controller. However I couldn’t get it to work. So Instead I spawned an emitter regularly at the target position.

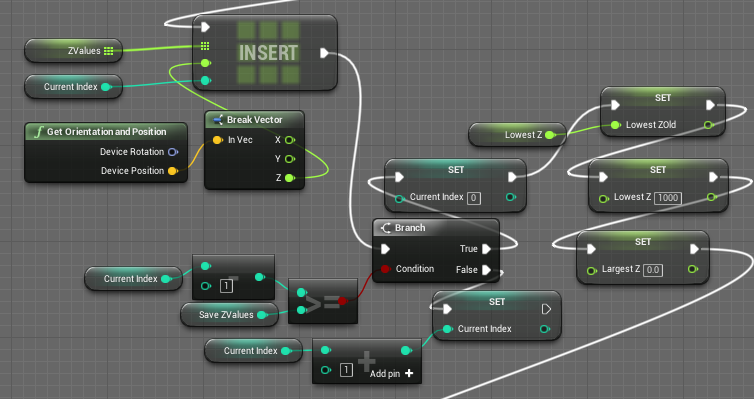
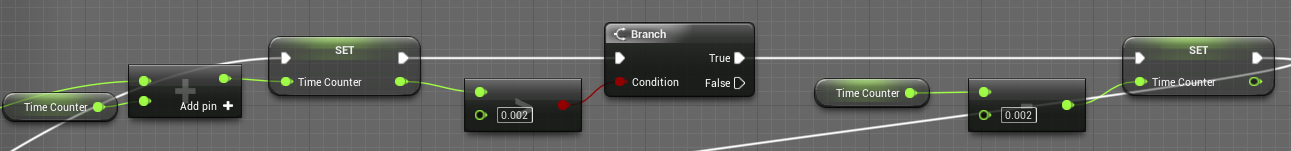
In the Jumping we need to measure zValues in certain time intervals.

Figure 13 - Measuring Time

Figure 14 - Creating an array of Z Values

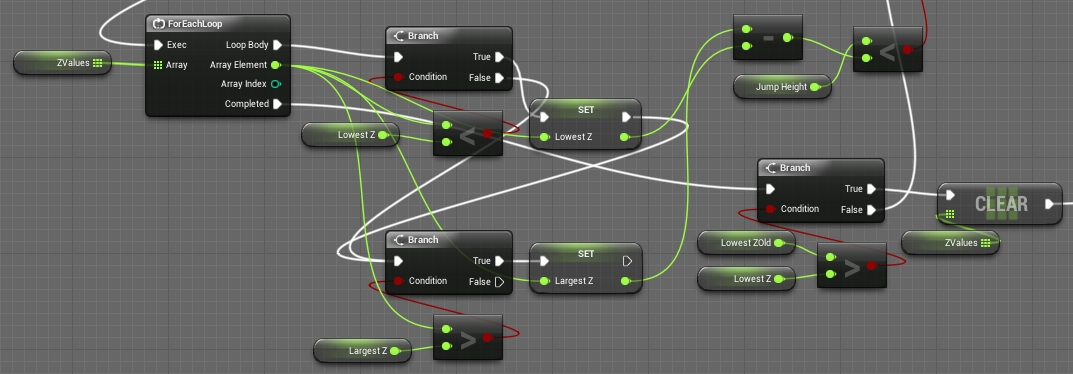
We then see if the the current Index is at that the highest point and if it’s so we reset the index back to the starting point, we save the LowestZ Variable, that we found in earlier steps and reset Lowest Z and Largest Z.

Figure 15 - Iterating and Evaluating

We now iterate over the ZValues and find ther largest and the lowest value, if there is a lower value found than the Lowest Z that is saved the array gets cleared, because we only want to jump when the user is going in positive Z direction. If that’s not the case we compare the jump height to the difference between the lowest and largest Z.

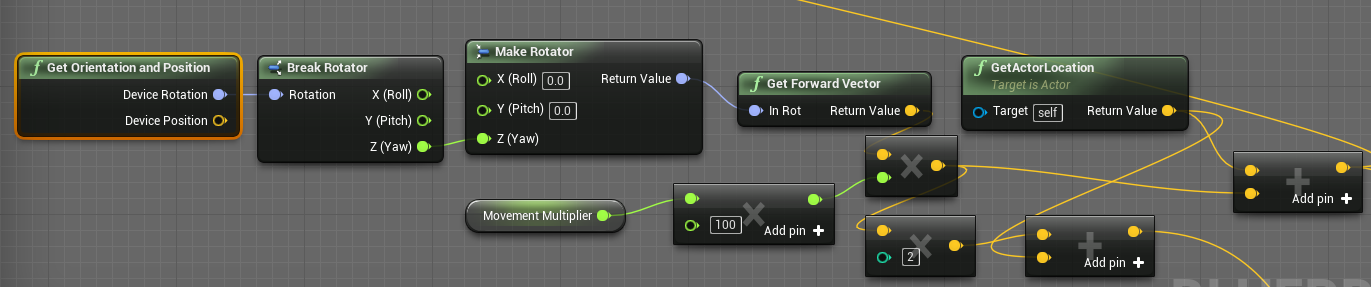
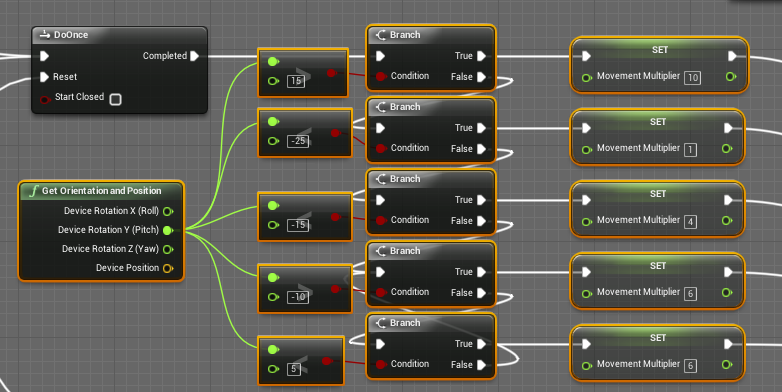
If the jump height is smaller than the detected difference, then the user jumps forward.

Figure 16 - Multiplier dependent on HMD Pitch

Figure 17 - Jumping Location Calculation

Figure 18 - Jump Height Detection

If the jump height branch doesn’t evaluate to true, then the multiplier is set by checking the value of the HMD (Pitch), this is if you look close to your shoes or far to the horizon.

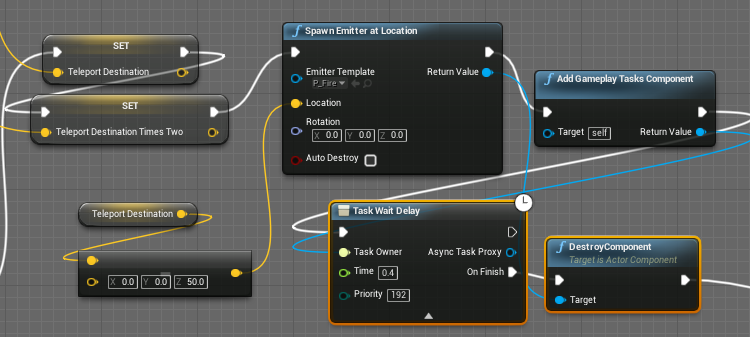
In the end an emitter is spawned at the teleport location, this is refreshed only all 0.4 seconds.

Figure 19 - Spawning a Target Indicator

## Combining the navigation methods

Nope see in further

# Testing (BOTH)

## Introduction (Dominic)

Introduction to the testing

Anzahl testpersonen

what has been tested, how has it been tested, etc

* SURVEY
* Testing procedure google doc

## 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 tested navigation methods. They had as much time at disposal as they needed to feel that they know how the navigation method works. We expected them to take one to two minutes to get the feeling for the method

* DIAGRAMME

Results

## Pick & place

What tested, what expected

* DIAGRAMME

Results

## Jump’n’Run

What tested, what expected

* DIAGRAMME

Results

## Ease of Use

What tested, what expected

* DIAGRAMME

Results

## Problems during testing

* E.g. the ladder (or other objects) was always visible
* Button not easy usable
* etc

# Conclusion (BOTH)

## Introduction

Intro to conclusion

## Insights

Welche Erkenntnisse haben wir gemacht?

## Suggestions

Schlussfolgerung

Konzept Suggestions which NavMet where to use

Entwicklungsprozess

# Further Steps (Marcel)

## Introduction (Dominic)

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, s.t. 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.

## Teleportation Idea of Simon Marcin

Teleportation with turning in the virtual world, when being to close to the chaperone in the real world.

WORK TO BE DONE HERE!

## Deceleration when close to walls and obstacles.

Deceleration and other Feedback when being to near to virtual walls can be used, to maybe create an immersive feeling that there is a wall. At this very point of the implementation the user is shifted back to another position when colliding with a wall, these effects should be minimized as much as possible when, player – wall collision is enabled.

# 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

Lessons Learned Dominic

### Marcel Groux

I enjoyed learning Unreal Engine a lot. It was nice to get used to program with the HTC Vive and Unreal Engine 4, since there isn't that much documentation around.

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

The workflow of putting the Headset on and testing some movement 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 basestation I ended up reloading older firmware to the basestation,

because one of the two 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 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 where 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 reevaluated the situation and decided to slow the process down at least at the beginning, so we could've helped out eachother.

This was a big error. Deadlines that were missed meanwhile the project week, weren't right away reevaluated. In retrospect, we should have talked right away with eachother.

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, such that we could test in the same week when we were ready.

In future we should fix dates and invite people earlier, such that the timeslots that 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 stress during the last few weeks of the semester.

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

## Collaboration

### Team Internal Collaboration

Due to working together in the projects 1 & 2 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 ended up dividing 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.

# Index of Literature (Marcel)

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

Index of all figures that will be in the text.

[Figure 1 - Walking in place concept draft 15](#_Toc472005774)

[Figure 2 - Scaled walking concept draft 16](#_Toc472005775)

[Figure 3 - Walking by leaning concept draft 17](#_Toc472005776)

[Figure 4 - Jumping concept draft 17](#_Toc472005777)

1. Attachment (Dominic)
   1. Project Agreement

Project Agreement?

* 1. Test Survey

Test Survey, with analysable results? -> Excel sheet

Google sheet evaluation

* 1. Attachment 3

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