Bachelor Thesis

The feasibility of low-cost Virtual Reality motion tracking for rowing technique analysis

Koen Vogel

Saturday 15th February, 2020

Supervisor: Dr. Ir Robby van Delden
Critical Observer: Dr. Randy Klaassen

bachelor
Creative Technology

University of Twente

Abstract

Injuries are very common among beginning rowers. One of the main reasons for these injuries, is rowers not using the proper technique, which can induce strain in body parts like the lower back. Automated methods for correcting faulty rowing technique are often expensive or have large delays. Therefore, this research proposes a novel method, using Virtual Reality (VR) technology to correct rowing technique, as well as to improve the immersion of indoor rowing.

The proposed system consists of an ergometer with a set of low-cost VR motion tracking devices attached, which are connected to a PC and a Htc Vive VR headset with a Head-Mounted Display (HMD). The motion trackers record the movement of the user, after which the data is analysed in real-time. Feedback is given using the HMD, in the form of barchart-like feedback system. To improve on the user's immersion, a visually pleasing virtual environment is created, where a virtual representation of the user can row on a river. The humanoid character resembles the user and replicates the user's movement using body-matching.

Extensive user tests were conducted with a sample size of 20 participants, where a group using the VR installation is compared with a control group using traditional feedback methods to improve on their technique. The results show that there is no significant different in rowing technique improvement between the two groups, and slight improvements on enjoyment and pressure/tension. The research concludes that the application of VR technology on indoor rowing is promising, since the tracking and analysis suffices to show improvement in technique over time, as well as improved immersion using the HMD. However, more research on the topic is recommended to draw more conclusive results.

Acknowledgements

First I would like to thank my supervisor Robby van Delden and my critical observer Randy Klaassen for their supervision and continuous help throughout this research. Secondly, I want to thank Dees Postma for his expertise and input in the ideation of this report. In addition, I would like to thank Richard Loos his rowing expertise and helping me find a research direction for this report. I also want to thank Jan Lammers from RP3 dynamics, for providing the ergometer used in implementation. On top of that I want to thank Label305, specifically Joris Blaak and Niek Haarman, for building and providing libraries used to process and analyse sensor data from the ergometer. Furthermore, I want to thank Abe Winters and Stijn Berendse, from rowing association Euros, who helped inexperienced me find my way through the world of rowing, and provided valuable feedback. Finally, I want to extends my thanks to all subjects who participated in the user test, helping me evaluate the proposed solution.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

Injuries can be very common in the sport of rowing. 32 - 51% of rowing athletes experience strain in the lumbar spine specifically [1], which only get worse with age [2]. Although overuse and intense rowing training schedules can be a cause, the root of the problem seems to be beginner level rowers not using the correct rowing technique [3], because improper technique can cause unnecessary strain on the athlete's body.

Despite many experts disagreeing on an exact technique, there is still a relatively universal standard to be found for the best rowing technique practice. This includes, but is not limited to, the correct order of muscle exertion, height of the oar, angle of the rower's back, timing and consistency. Following this form optimizes the rower's effective power output and minimizes strain. This technique varies however, not only per boat model but especially when comparing rowing on the water with rowing on an indoor training machine (ergometer).

Despite this difference, indoor rowing is an important part of the rowing sport because of two main reasons. The first being inconsistent conditions like weather, boat- or teammate availability and temperature, which often do not allow for outdoor rowing. And the second is that ergometer training allows for much more coaching possibilities, as the coach can give personal attention and correct the training rower's technique. This all results in rowing clubs, teams and associations often having large quantities of ergometers, which are regularly used.

1.2 Goal

In this field of sports exercises, an increasingly popular research topic is Virtual Reality (VR) [4], especially solutions with a Head-mounted display (HMD). This technology allows the user to completely immerse themselves in a virtual 3D environment, with intuitive interaction methods such as motion tracked controllers. In the specific context of indoor rowing, VR has already been used for commercial immersion and engagement purposes, for use in a casual environment, but there is a lack academic evaluation. Therefore the benefit VR could bring to indoor rowing will be explored, on two different aspects. First is the technical aspect of rowing, namely the rowing form or technique. The main objective of this report is to use VR and its motion sensing capabilities to analyze the rowing technique of indoor rowing athletes in real time and provide corrective feedback, in order to stimulate technique improvements and thereby decreasing the risk of injuries.

Neumann et al[4] also suggest that one of the core features of VR is increased immersion, which might impact the performance in sports. Ijselsteijn et al[5] add to this that the level of immersion is closely correlated to the motivation in sports and physical activity. Therefore, as a secondary goal, the possible immersion improvements of VR in indoor rowing will be researched by evaluating the impact of an immersive virtual environment.

1.3 Research questions

From this set of challenges, a few concrete research questions can be formulated to set out a guideline for the following research.

Main question

- How can Virtual Reality improve the current state of indoor rowing?

Sub questions

- Can rowing technique be analysed and corrected using VR motion trackers and digital feedback?
- Can the immersion of indoor rowing be improved using a head-mounted display?

2 State of the art

2.1 Related work

This section will give an overview of existing projects and commercial products that are related to this report's goal. After summarizing the findings with their advantages and disadvantages, useful information can be extracted on what has been done in this area, and what possibilities for improvement still exist.

While literature was consulted, the results below are found using a commodity search engine, as these appear to form a more detailed overview.

2.1.1 Holofit - holodia.com

Holofit is a Virtual Reality game collection designed by Holodia, which runs on an HTC vive, a commercially available VR headset. It connects to a wide variety of bicycle and rowing ergometers using the Fitness Machine Service Bluetooth protocol, in order to use the data from their power sensor. The company started its product as a kickstarter project and is now commercially selling a few package options including just the software, or a bundle including the HTC vive and a PC capable of running the headset.

The provided software depends on what kind of ergometer is connected, but for the purpose of this research, the games designed for rowing ergometers will be discussed. It consists of a 3D representation of the user in a boat, which responds to power input from the ergometer, moving the boat along, see figure 1. An interesting fact is that the user is moving in the direction he/she is facing, which is the complete opposite of how a regular rowing boat/scull works. This is a clear indication Holofit more focussed on entertainment rather than accuracy. Several high detail virtual environments are provided to optimize user satisfaction, but there are a few performance tracking features as well.[1]

The measured speed, in combination with time, is used to motivate the user to perform better, by comparing these 'scores' to previous entries and other users, and visualize these in VR. Another way holofit is trying to stimulate performance is multiplayer options, where multiple people can row next to each other in VR and thus are motivated to perform better by appealing to their competitive spirit.

All together, holofit appears to be an entertainment focussed, social, rowing themed VR game which provides visually pleasing environments and stimulates users to perform better by using gamification of performance.

[1] https://www.holodia.com/#single/0 - *video of the holofit in action*

Figure 1: A screenshot from the holofit software, source: holodia.com

2.1.2 The Quiske System - rowingperformance.com

The Quiske system, by the Finnish company Quiske, is a solution consisting of a mobile app and a sensor device, the Quiske Pod. This device, when attached to the seat or oar in a rowing boat, tracks and analyzes rowing performance in real time. The device, seen in figure 2, contains an array of sensors and communication hardware. While not explained on their website, this sensor array likely contains a gyroscope, accelerometer and bluetooth transceiver, which record the movement and orientation of the pod and send this data to a connected smartphone with their Quiske app, see figure 3, installed. The app, which is free, can work without a connected pod, but in that case the gathered data will be that of the smartphone's built-in sensors, which is less accurate

The feedback provided by the Quiske app is displayed in the form of handle or seat speed curves, evaluated and given a technique score from 0 to 100. Together with elaborate instructions for execution order and style of rowing technique in general, this is positioned by the company as a complete replacement for a professional coach.

The Quiske system seems to be very potent due to its ease of use, because the effort required to integrate the system in a normal rowing session is very low. Its ability to replace a coach seems questionable however, due to the systems low accuracy.

2.1.3 Row Analysis - rowanalysis.eu

Row Analysis is a Dutch company which aims to provide a service to improve rowing technique in a unique data-driven approach. Once purchased, customers will receive a marker set which attaches to specific parts of their body and a rowing ergometer, namely the feet, handle, seat and shoulder. They are instructed to film themselves, using any camera, completing ten rowing strokes on an ergometer, while all markers are visible. After sending this recording to

Figure 2: The Quiske pod; source: rowingperformance.com

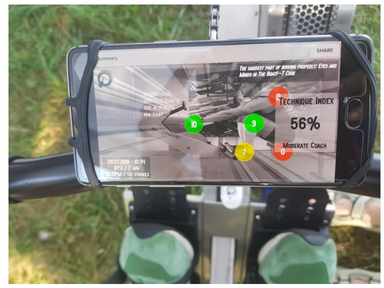

Figure 3: The Quiske app; source: rowingperformance.com

the company, it will be analyzed and a detailed report will be sent back within a few days, containing results of the analysis and recommendations about rowing pace and technique.

Their process, while not entirely public, is partly explained on their website and works as follows. First, the positions of the reference markers are extracted from the provided video using computer vision, which is an automated tool to detect features in imagery. Then the movement, per reference point, is calculated using motion tracking, which provides cohesive data for the company to use in order to analyze technique. This is presumably done partly with automated systems, as they state that “Benchmarking is done based on the information in our reference database”.

Automated or not, the legitimacy of this reference database is very important because it forms the basis of their findings. Despite the terms of Row Analysis stating that they do not guarantee the quality of the benchmark data, their site does provide references and substantiation for their calculations. For instance, the company’s founder and several affiliates are former (professional) rowers and/or professional coaches.

In conclusion, Row Analysis is a company which uses their rowing expertise to provide customers with a unique, easily accessible way to evaluate their technique, using state of the art technology.

2.1.4 Conclusion

While not a complete collection of all commercially available related work, the discussed products give a clear overview of the different perspectives in the field. The Holofit is an entertainment focussed product which uses VR and gamification to increase engagement, while Quiske and Row Analysis have a clear focus on delivering value for rowers with higher skill levels in the form of technique and performance analysis.

Figure 4: A motion tracked rower, source: rowanalysis.eu

The fields of gamification in sports and automated performance analysis both show promise for the future but there appears to be a lack of solutions in between, in the form of gamified technique analysis in real time. Presumably one of the reasons for this is a lack of low-cost accurate tracking solutions, since the discussed solutions only use a very limited amount of sensors.

2.2 Rowing Technique

For the proposed system to give accurate feedback, the measured motion and power data has to be analyzed and compared to a 'correct' example. This section sets out to find a conclusive model for the system to use as a baseline to compare to, with a few criteria; First, this model will be solely applicable for the ergometer, not on actual rowing on the water, because of the large difference and time restraints. Secondly, the model has to be universally applicable, meaning that the definition is broad enough to not incite discussion among experts, of which there is much in high level technique comparisons. Due to this aspect being so broad, the focus of the definition will be on beginner to intermediate skill level rowers, and their common mistakes. Intermediate level rowers will be observed and coaches will be interviewed in order to form a base of understanding.

The sources for the technique model consist of experts. Both interviews and observations on location have been conducted, where experts in the field of rowing, like coaches, provided the information the following section has been based on. All interviews and observations can be found in appendix A.

2.2.1 Rowing stroke

The structure of this model will use one complete ergometer rowing stroke as reference. The start and end of the stroke can be defined as the position, compared to every other point during the stroke, where the back of the rower is at its largest angle, their legs are maximally retracted, knees at the highest position, and the ergometer handle maximally retracted.

The technique can be divided into four phases, as seen below.

- the catch
- the drive
- the finish
- the recovery

After phase 1, the grabbing of the handle (or inserting the oars in the water) the second phase is defined by a drive in the quadriceps for around 80% of all required force, followed shortly after by the lower back and arm retraction. While a distinction in order is important, for optimal effect on the water, a smooth transition between these muscle activations is crucial. A force curve, like that produced by most ergometers, show these hesitation or hiccups in the form of irregularities in the curve, as seen in figure 5, and thus is a good solution to analyze an athlete's force distribution. The third phase describes the ending of a stroke, when the back of the athlete is at its maximum declination. A rowing stroke ends with the recovery phase.

This recovery phase is, arguably, the most important phase in the rowing strokes, because this is where the most common mistakes are made. When ending the finish phase, the order of execution starts with the extension of the

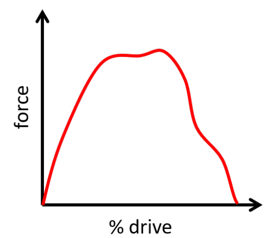

Figure 5: Illustration of a force curve during the drive phase

arms, followed by incline of the lower back and at last by the retraction of the legs, which brings the distance between the seat and flywheel to a minimum. This order of execution is the most optimal way to keep the handle of the ergometer at a constant height, which is crucial because the oars should not hit the water during the recovery in a real rowing situation, as this would drastically impact the boats speed and stability.

2.2.2 Common mistakes

The most common mistake is improper execution order of muscles in the recovery phase. This happens because the movement required to keep the handle at a constant height is very unintuitive for inexperienced rowers. A widely used method in improving on this, is dividing this order in four points, so beginners have a clear reference to compare to. The first three points can be seen in figure 6, while the last point is the rest position, the start position of the entire stroke.

The use of the ergometer handle or rowing oar is also important, as coaches see many beginners squeeze the handle too hard, or move their wrists too much, which puts unnecessary strain on the body.

Furthermore, many rowers fail to form a hollow back during the drive and finish of the stroke, which decrease the efficiency and thus the performance of the rower.

Lastly, inexperienced rowers are often seen pushing their core away, bending their body, instead of push their whole upper body to the back. This is because they do not use their abdominal muscles enough, resulting in wasted energy and possible strain in the lower back.

Figure 6: Steps 1, 2 and 3 in the recovery phase of the rowing stroke

2.3 Virtual Reality in sports - Literature review

2.3.1 Introduction

Before starting this project, more background research is needed on VR, specifically VR technologies which utilise a HMD. This Literature research will therefore review academic sources on two topics. First the current state of research and application of VR in different sports will be examined, by asking the question: 'What is the current state of VR research and application in sports?' Then the biggest hurdle in the way of VR growth will be described and discussed by asking 'What are the causes and implications of motion sickness in Virtual Reality?'

2.3.2 Sports and Fitness

A relatively new and upcoming use case for VR is as a tool for sports and workout sessions. In the early days of VR, the technology was mostly used as a demo for immersion purposes, with relatively passive environments and low interaction value, but although being experimented with for a while, it was recently when fitness focused VR has exploded in size, thanks to a few popular applications. This section will go over the trends and influences of fitness for VR and discuss its benefits and problems.

One of the key methods of successfully integrating VR and sports is gamification. Deterding et al[6] put forward a working definition of gamification in the form of "Gamification is the use of game design elements in non-game contexts." [6, p.2]. One way to apply this up and coming concept is influencing a user's behaviour by appealing to their 'competitive spirit' [7]. In their VR application build for fitness, Tuveri et al [8] found that, with the help of gamification, the addition of VR significantly increased the enjoyment of the users, and therefore created a set of guidelines for this type of 'fitmersive' games. Together with many other found 'exergames', a gamified application with the goal of providing exercise benefits to its users, this shows that VR is indeed an effective method of motivating people to exercise. Gamification can also come from another perspective however, as Yoo et al [9] described, traditional, entertainment focused VR games can provide a lot of exercise value as well. Because of the workings of VR interaction methods like positional and rotational tracking controllers, physical movement is an innate function of many VR games existing on the market today, which is why Yoo et al [9] cite that even general use can have a positive effect on user's physical well-being.

Another way VR is used in a fitness context is analysis. With the market for VR rapidly growing, the technology is getting cheaper and better. This means that the use of VR sensors for tracking the position and rotation for both head and hands, might be more feasible than commercial or industrial class IMU's or other sensors in certain contexts [10]. Together with the use of gamification, the use of VR to analyse and evaluate the performance of fitness and sports is a growing field. Hulsmann et al [11] used VR to successfully track the performance of squats and Tai Chi pushes and provide the user with real time feedback, using their proposed solution. Similar methods have been used to provide feedback in different types of training, all with positive results and user satisfaction [12, 13], this clearly shows that VR is indeed an effective method of performance analysis in sports and fitness.

However, there are disadvantages of using VR in sports, especially if used as a substitute for real physical activity. Varela-Aldas et al [14] found that the heart rate, a clear indicator of physical effort, of users in their study was actually significantly lower when exposed to a VR exergame. They therefore conclude that, despite positive user satisfaction, VR cannot be used as a complete substitution for real physical activity.

2.3.3 Motion Sickness

One of the biggest downsides associated with VR technology is motion sickness. Especially for first time users, VR games and applications with lots of movement are shown to induce the side-effect. In this section, definitions and possible causes of VR induced motion sickness will be discussed, with their health implications and possible solutions. Although there are several different definitions and names for this concept, like Cyber Sickness and Visually Induced Motion Sickness, a few universally agreed upon causes can be defined.

The most widely known cause is that of discontinuity between visual and inertial perception, also called Sensory Conflict Theory [15]. The user will perceive motion and rapidly changing visuals and information through a Virtual Environment (VE) but the body does not is in a still position and does not perceive any of this motion. This causes a disconnect of several sensory inputs and thus induces motion sickness.

A second relatively well known theory is that of Postural Instability [16]. It states that motion sickness can be induced in a user by putting them in an unknown environments and situations, in which a user needs to adapt their posture to stabilize. Just like with sea sickness, experienced users have already adapted to this feeling, but new users have problems adapting to the new way of maintaining postural stability. Chardonnet et al [17] as cited by Chattha and Shah [18] confirms that the postural instability theory applies to VR specifically very well, stating that it is a large factor contributing to VR induced motion sickness.

To evaluate the probability of motion sickness occurring in the use of VR, a study conducted by Regan [19], found that 61% of 150 subjects experienced motion sickness in a 20 minutes long period of VR immersion, with 5% of subjects withdrawing due to the severity of their symptoms. She concludes that anti-motion sickness drugs are a successful way of mitigating this. Chattha and Shah [18] found similar results with their study and add that, in their findings, female participants were significantly more prone to motion sickness than men.

Despite this high rate of motion sickness, the current state can be improved. Zaidi and Male [20] concluded that although motion sickness or cyber sickness can be overcome by the improvement of the technology itself, which is already on its way, the most important factor of motion sickness relief is user acceptance, e.g. getting used to it through prolonged use.

2.3.4 Conclusion & Discussion

In the first section the current state of VR in sports is discussed, where two main applications were identified; The analysis and tracking of performance with the help of VR sensors and exergames, gamified experiences to increase user satisfaction. The downside being that VR fitness applications do not physically stimulate the user as in a real workout. Despite this, both applications are up and coming fields with promise for the future.

Due to the limitations of this review however, only the two most important applications were discussed. For further research, review of more applications in sports is recommended, with a specific recommendation for research into the use of VR to users with physical disabilities and its implications in sports, which could be a very interesting field of research.

In the second and last section, cyber sickness, or motion sickness for VR, is discussed. Two causes of the phenomenon were identified; Sensory Conflict Theory, which is linked to the disconnect of several sensory inputs, and Postural Instability, which is the ability of the user to adapt their posture to new and virtual environments. Both are valid concerns, and contribute to the large effect it has on the large scale adoption and growth of VR technology in general. There is hope for the future though, with both user acceptance and improving technology leading to a decrease in motion sickness over time.

Both of the reviewed causes are valid but scientific substantiation could be improved. Used sources [15, 16, 19] for the theories and the application of which in VR are outdated, and because of the rapid change and growth of VR technology, need to be improved. Therefore it is recommended to review and research the causes of motion sickness with modern VR hardware.

3 Methods and techniques

This section will set out to explain the general structure of the project. This is done by describing the different methods and techniques used in the process of the research, namely the Creative Technology Design Process. The section will conclude with an outline, describing the next chapters of the report.

3.1 The Creative Technology Design Process

The Creative Technology Design Process (CTDP) is a design process widely used in the Bachelor study Creative Technology. CTDP is based on two models of design practice[21]; The Divergence and Convergence model and the spiral model. The Divergence and Convergence model[22], defines two phases in the research process. A diverging phase, which explores the subject and generate many concepts, and a converging phase, which narrows down the research to a single solution. On the other hand, the spiral model is implemented, which allows for more iterative design process with steps that do not follow an iterative order.

As seen in Figure 7, the traditional implementation of the CTDP consists out of four phases. The ideation, specification, realisation and evaluation. The ideation phase describes an exploration of ideas and the designing of possible solutions. The specification phase further specifies the best concept(s) from the ideation phase, by defining requirements. Small evaluations of the concept and further iteration also fall into this phase. The realisation is all about implementing the specified concept, by building the solution according to the requirements. The last phase is the evaluation, in which the realised solution is tested and evaluated.

3.2 Integration

For application to this research, the CTDP is slightly adapted to fit our needs. As the project does not focus much on the specification phase, the final concept will be narrowed down and described inside the ideation phase. All implemented phases are described as follows.

Ideation

In the first phase, the ideation, of this report, will explore a range of concepts which apply VR technology to the problem. The sections starts with the techniques used, and will go on to narrow down the problem. Some concepts are then designed, which are iterated upon and narrowed down to a single solution, consisting of the different hardware and software parts needed to solve the problem set out in chapter 1.

Realisation

The second phase is called the realisation. There, methods and technologies used are explained and the different aspects of the final prototype are described, like software and hardware. Additionally, the process of implementing the chosen solution is described, along with problems found during the building process.

Evaluation

The final research phase is the evaluation. This is where the prototype constructed in the realisation is evaluated on its functionality and effectiveness, in this case with user tests. The methodology and procedure will be explained, and the results gathered will be visualised, after which statistical analysis can be conducted.

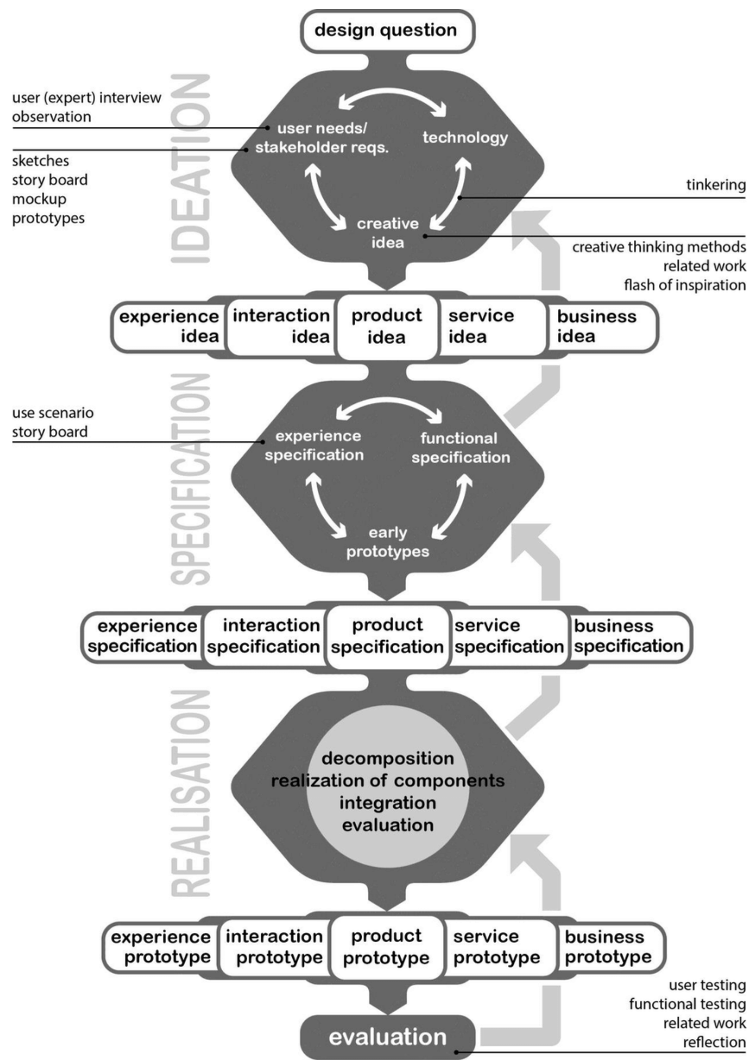

Figure 7: Overview of the Creative Technology Design Process

4 Ideation

In this section, the research questions in chapter 1 will be narrowed down to a usable list of criteria for a prototype, by ideating and discussing concepts and defining a set of specifications for installation.

4.1 Methodology

From the research questions of this report, can be understood that a prototype has to be made which incorporates VR with a rowing ergometer, use motion tracking to analyse the rowing technique of its users, and visualise correcting feedback in an intuitive way. To narrow this down, the ideation of the prototype will consist of four aspects; Tracking, Technique analysis, Feedback and Immersion. Each following subsection will start by defining criteria, if possible with the help of experts and stakeholders, after which different concepts and solutions are ideated and discussed. Each subsection will be concluded with a chosen concept, which will be specified in the next chapter.

4.2 Tracking

For technique analysis to be done, data is required of the human posture. To gather this data, some form of motion tracking will be needed. This subsection will try to find the most accurate, low-cost tracking hardware, which is sufficient for human gait analysis.

Sensor types

According to Gouwanda and Senanayake [23], magnetic tracking systems and optical motion capture systems, in the context of human gait analysis, are both sub-optical solutions due to line-of-sight restrictions and signal distortion respectively. They state that, due to the advancements in Micro Electrical Mechanical System (MEMS), Accelerometers, Gyroscopes, magnetic sensors and combinations of these like inertial measurement units are a viable alternative. This is because of their small size, lightweight, low cost and low power consumption features make them easy to mount on the human body. Aminian and Najafi [24] confirm this and add that the possibilities of real-time analysis are promising, because of the short processing times in comparison with conventional tracking systems. They also evaluated their system with generally positive results.

These types of motion sensors were also tested and evaluated in human gait analysis by Sun and Sakai [25], who concluded that they give very accurate results, given an error compensation algorithm is applied.

Sensor positions

With these sensors, the position and rotation of points on the human body can be accurately tracked in real-time. Now, an array of tracking locations on the human body has to be found which data can accurately represent a complete human posture. Using conventional trackers, a lot of these tracking points were needed, because they only track position, meaning rotations were calculated using multiple tracking points [26]. Using Micro Sensor Motion Capture (MMocap), a MEMS technology which combines gyroscope, accelerometer and magnetic sensor data, far fewer tracking points are needed for a complete reconstruction. After small amounts of on-board processing, these sensors send data streams in the form of quaternions and x, y, z coordinates for rotation and position data respectively.

Tao et al[27] set out to build a tracking model with this data. They recommend setting up a human rig with 16 sensors in total, with three per limb and four divided over the torso and head. Using this setup they can accurately reconstruct a human posture with the provided data. With the specific focus on the lower limbs, Ahmadi [28] uses an identical configuration of three IMU sensors per leg, which is enough to represent a human posture, although it has to be said that both of these approaches are tailored to gait analysis specifically.

Integration

For the application in a prototype, less tracking points are required, as a calculated estimation of joint positions can be made. These are far less accurate, but the scope of the project has to be considered. Due to available time, work and costs, a compromise must be made and only the most necessary trackers have to be kept. Since joints in the middle of a chain can be estimated by calculation, using techniques like inverse kinematics[29], the hands and feet of users are one of the most important to track right. Here are some shortcuts to be made though. We can assume that the user has their feet in the ergometer footrests and their hands holding the ergometer handle at all times, which is a fair assumption as motion tracking is only relevant during rowing strokes. This, in turn, means that the distances between hands and between feet should stay the same. Tracking the ergometer handle and footrests are therefore sufficient to get an accurate representation of limb posture. The same principle can be used to limit the number of trackers on the other side of the limbs, by tracking points between the shoulders, for the arms, and on the lower back, for the legs.

For the choice between sensor types, it is clear an IMU is a good choice in the context of human motion analysis. Despite there being a lot of different sensor models, the choice of sensor is again constrained by the scope of the project. For this reason, integrated sensors of VR systems have to be considered. The tracking system of the HTC Vive, one of the most common VR systems, is shown to be reasonably accurate [30], with a high update rate of 120Hz and low latency of 22ms. This tracking system consists of an IMU in combination with infra-red tracking sensors to provide absolute positional and rotational data. Despite the slight decrease in accuracy compared to alternative IMU's, this system has several advantages. Because of the accompanying firmware and software development kit (SDK), the error compensation for brief losses of tracking is good. On top of this, the SDK and integration with game engines make the development of VR implementation very easy and time-effective. For these reasons, the HTC Vive system is chosen for fitting the wanted immersion in the virtual environment as well as the motion tracking for technique analysis.

4.3 Error detection

With this effective hardware, a model can be constructed with which a configuration of trackers can analyse rowing technique. As seen in section 2.2.2, there are many aspects to rowing technique that the target users, beginning rowers, have often troubles with executing correctly. This section will narrow down the rowing technique which the prototype is aimed at analysing and correcting, as well as establishing a configuration of trackers needed to accomplish this.

From the previously mentioned list of common rowing mistakes, the most relevant criteria is the prevalence of the mistake. According to the expert sources, the steps in the recovery phase of the rowing stroke are the most essential and most common mistake to correct, which makes it the most relevant option for implementing in the technique correction model. Of course, it is possible to implement the analysis of a wide variety of technique aspects and mistakes but there is chosen for this single one. That is because this report is aiming to provide a proof of concept; a large and detailed technique correction model is not very relevant and thus falls out of the scope of the project. If VR appears to indeed help with technique correction, an extension of this model can be contemplated.

With the chosen technique aspect, a more detailed model can be constructed. As explained in section 2.2.2, the separation of the rowing stroke into steps mostly concerns the division of the stroke movements per limb/muscle group, namely the arms, back and legs in the recovery part of the stroke. The timing and duration of these different limb/muscle group movements, therefore, need to be detected using motion tracking. With the chosen technology, the Htc Vive, this has to be done with the external motion tracking devices, the *Vive Trackers*. These are commercially available consumer-level devices, which can be bought separately for around €125, as of the time of writing, and integrates seamlessly into the steamVR API.

To accurately track the timing of arm, back and leg movements, these trackers need to be distributed among the ergometer and user. As discussed in section 4.2, there are some shortcuts to be made, such as using the same trackers for both legs. On top of this, because of the low-cost nature of the challenge, a small number of trackers would be ideal. Therefore the detection of movement is chosen to be limited to two trackers per limb, at both ends, with which the change in distance can be seen as the movement. The overlap between trackers used for different limbs is also desirable, e.g. using the same tracker for measuring leg movement as well as back movement. The prototype should be constructed with these requirements in mind.

4.4 Feedback

When designing a feedback system for the analysed technique, an important requirement which has to be contemplated, is the cognitive load on the user. Isbister and Mueller [31], in their guidelines for designing movement-based games, state that "Moving can demand a lot of mental attention, creating high “cognitive load,” especially when learning new movements, so do not overload the player with too much feedback.". Following these guidelines, the feedback system in question should be as simple and intuitive to use as possible.

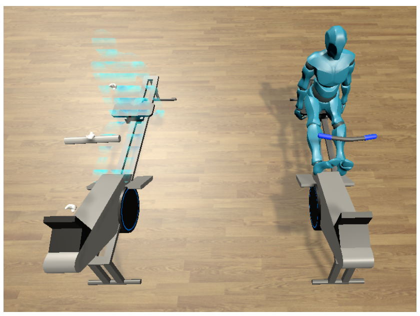

Figure 8: First concept of the feedback system

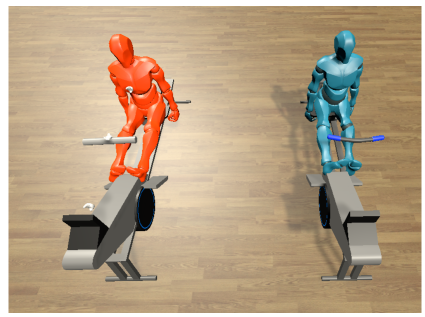

Figure 9: Second concept of the feedback system

The first two refined concepts, which attempt to utilise a low cognitive load on the user, can be seen in Figures 8 and 9, and work as follows. The character embodying the user, seen on the right in both figures, will be analysed. The best rowing strokes, as determined by the technique analysis model in section 4.3, and stored in volatile memory. During the run-time of the simulation, recorded strokes would be selected and visualised in a ghost-like second character model, as seen in the left of both figures. This is done by applying the recorded position of all tracked points on the installation, relative to an origin, or root of the character model, to the ghost character model, in real-time. An additional ideated option was that experts would be able to record a ’good’ rowing stroke, stored in non-volatile memory, instead of selecting only user-recorded strokes.

The first concept was textured using a hologram-like shader[2] in Unity’s *shader graph*. After finding that the shader was hard to distinguish from the background, and thus required too much effort to interpret, the second concept was made by modifying the colour of the first one. A short evaluation on the first two concepts was done with an expert in this field, sports and movement scientist *Dees Postma* from the University of Twente. This showed that the concepts were unclear in interpretation and required too much explanation or instruction to use effectively.

With this in mind, a few more concepts are ideated in a brainstorming session. This is done together with Dees Postma. In Figures 10 and 11 the third and

[2]tutorial by *https://www.youtube.com/watch?v=KGGB5LFEejg&t=39s*

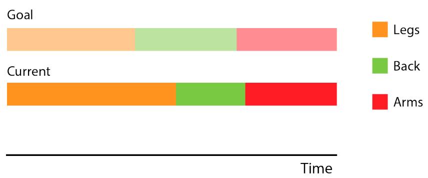

Figure 10: Third concept of the feedback system

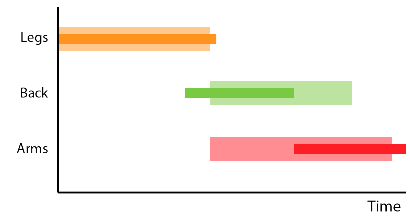

Figure 11: Fourth concept of the feedback system

fourth designed concepts can be seen, where the more transparent bars indicate the goal configuration, which the user is stimulated to replicate with the opaque bars, which react to the analysed technique of the user each rowing stroke.

One of the most important considerations in designing the second and final prototype, seen in figure 12, was the ability to display overlap between the movements. The earlier concepts only allow for bars that start when the previous one ends, while an important part of rowing technique is the amount of overlap in the limb movements.

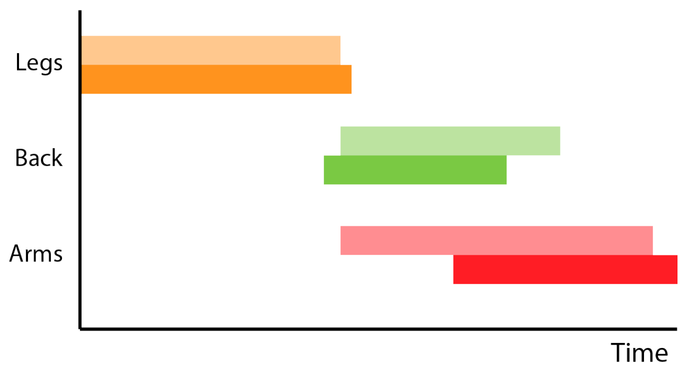

Figure 12: Final concept of the feedback system

4.5 Immersion

As per the second subquestion of this report, one of the goals is to improve on the immersion of using an ergometer. According to Slater and Wilbur [32], a user's immersion in Virtual Environments (VE's) is closely related to user presence and can be improved by creating a rich environment and, most importantly, a high degree of body matching. This last one is a technique where the posture of the user is being copied to the virtual environment as accurately as possible.

The ways comparable systems have implemented immersion improving environments is by providing a visually pleasing virtual world to row in, with rivers and interesting terrain features, see chapter 2. Body matching, however, has rarely been implemented to a high degree. This is because common techniques used for this involve tracking user's movement with a collection of expensive sensors. As discussed in this chapter, however, this report's prototype will have to include several motion tracking sensors to analyse rowing technique. This means the implementation of common body matching techniques is feasible.

One of the most popular body-matching techniques is called Inverse Kinematics [29]. While traditional character animation technique, forward kinematics, work by rigging a character with a skeleton structure and manually positioning individual joints, Inverse Kinematics (IK) works in a completely different way. Instead of animating joints from parents to their child-joints, a child-joint is anchored to a position in 3D space and parent joints, up to a certain amount, are automatically animated to estimate a natural chain of positions to the child-joint, using a complex algorithm. For example, if the hand of a character were to be positioned to grab something, the elbow and shoulder joints would adapt to let the character's arm follow along. With this method, only a few tracking points are needed, at child-joints, to get a decently accurate estimate of full body posture. For this reason, IK is chosen to be implemented in the prototype.

Besides body matching, the environment must also be created to maximise immersion. Preferably with some interaction, as that is shown to increase user presence [32]. There are a few options to do this. First, the ergometer used could be modelled realistically. This has the advantage of increased user presence, as more parts of the real-world environment are extended to the virtual one, disadvantages might be that it would not be as interesting or engaging as alternatives. Another option is to model a completely different, visually pleasing river, with a rowing boat instead of the ergometer. The tracked movements of the ergometer handle could be translated to oars in the rowing boat and a visually interesting terrain can be constructed. This could increase the level of interaction and keep the environment visually interesting. Because interest and engagement are more relevant than realism, the second option is chosen. The last thing to consider is a possible side-effect. The virtual movement of the user, through a river, might induce some motion sickness, as there is a disassociation between real-world movement and virtual movement [18].

5 Realisation

After coming up with a concept for the prototype, this section goes on to describe the implementation process of the installation. Both software and hardware systems will be explained, along with the process and methods used.

5.1 Physical setup

The indoor rowing machine, or ergometer, used for the prototype, is the RP3 model t. It was provided by the company RP3 dynamics[3], a stakeholder of this research. Compared to a traditional style ergometer, like the widely used Concept 2[4], it has a few interesting advantages. First, instead of a static body/flywheel mounted on the base, the ergometer flywheel can freely move along the rails, just like the ergometer seat. On top of that, there is a damping system in one of the legs of the ergometer, which allows for slight vertical movement. These features aim to decrease the risk of injuries and provide a rowing experience which more accurately simulates a real rowing boat.

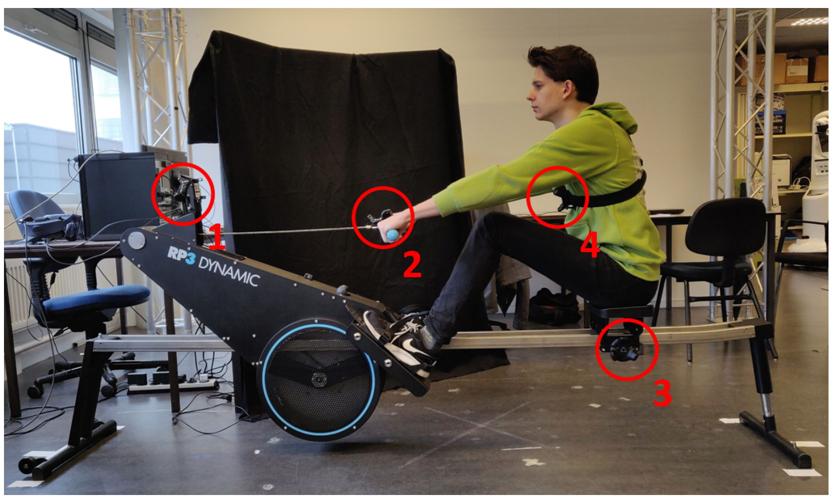

Figure 13: Locations of the motion trackers on the final prototype

The position of the motion trackers is important, as it influences the accuracy of both the technique analysis and the body-matching implementation. The final tracker configuration can be seen in Figure 13. Three of the motion tracking devices are mounted to the ergometer itself while one is attached to a chest-strap the user is required to wear. While the tracking accuracy of these devices is supposedly reasonably good[30], during implementation it was found that the tracking accuracy was very inconsistent, with short periods of significant

[3]https://www.rp3rowing.com/
[4]https://www.concept2.com/

Table 1: Movement limits of the tracking devices, per axis in 3D space

Tracker	Position	Can move in these axes	Can rotate around these axes
1	Flywheel	z	none
2	Handle	x, y and z	x, y and z
3	Seat	z	none
4	chest	y and z	x

inaccuracy. A reason for this *might* be the large number of trackable objects in a small space. To combat this, the axes of recorded movement and rotation were limited in software per tracker. This could be done because positions of the tracking devices in physical space is limited to the movement of the ergometer, e.g. the seat of the ergometer can only move in a single axis along the rails. This and the fact that the aspect of rowing technique the prototype analyses is very limited, means certain axes are irrelevant and can thus be ignored. An overview of movement limits for tracking devices can be seen in table 1, where the x-axis is perpendicular to the rail of the ergometer, y the vertical axis and z the axis parallel to the ergometer rail.

5.2 Algorithms

To start analysing the data coming from the motion trackers, the trackers are connected to the Unity software environment through the SteamVR API. This allows for the representation of all motion trackers as objects in a 3D virtual scene. When these trackers were calibrated and accessible, the position and rotation data for every tracker is recorded with a sample rate of *20Hz*, as seen in appendix B.1.1. With this data, the system detects when rowing strokes are completed and consequently analyses the rowing technique. This section describes the way stroke and technique detection is done in the prototype, based on the ideated technique correction model in chapter 4.

5.2.1 Stroke detection

First, is the real-time detection algorithm of rowing strokes, which should fire system-wide events at the end of every stroke, so the system can use all recorded data to analyse technique. For this, two methods were attempted. When the end of a stroke is detected, all recorded motion data is saved per stroke and the recovery phase is isolated.

The first method to detect the end of a rowing stroke is using a connection to the ergometer itself. The used *RP3 model t* ergometer, measures force information using a sensor in the flywheel, and through a Bluetooth signal or serial connection, it outputs raw sensor values. In a commercial context, this information is usually received by a phone or tablet running RP3's application, which visualises force data in graphs and has its own built-in stroke detection. To take advantage of their, tried and tested, stroke detection, I met with the company developing this app, *Label305*, a small stakeholder in this project.

They provided me with their proprietary library, responsible for stroke detection among other things. First, the raw data from the ergometer was imported using Python, Processing and the Arduino serial monitor. Then, attempts were made to process and receive this data with the provided library, using Java, binary sockets, and Networking libraries for C#. After failing several attempts, this method determined out of scope for this research and therefore rejected in favour of a simpler solution.

The second and final method used for stroke detection is a real-time algorithm written in C#, using the position data from the motion trackers. The algorithm can, partly, be seen in appendix B.1.2, and works as follows. It starts by calculating the velocity of all motion trackers, as well as the change in distance between several trackers. Per sampling frame, these velocities of the current frame, last frame and second-to-last frame are averaged and recorded as the final velocity value. This is to decrease the impact of outlier values due to tracking inaccuracy. The distance in on the z-axis, parallel to the ergometer rail, between trackers 1 and 2 are then used for stroke detection. When the change, or velocity, of this distance crosses zero, we can assume that the rowing stroke switched from the finish phase to the recovery phase, or from the recovery phase to the drive phase, depending on the slope of the velocity graph. Because the recovery phase contains the only relevant technique information, the motion data between these two points is recorded and used for technique detection.

5.2.2 Technique detection

When the system has isolated the recovery part of a stroke, it can start detecting the movement of the arms, back and legs, as discussed in chapter 4. It does this once per rowing stroke, right after the stroke has finished. This detection is different for the different body parts, and the trackers used can be seen in table 2. For the arms and legs, the change in distance between trackers is used, and the algorithm, also seen in appendix B.1.4, goes as follows.

Table 2: Metrics used for detecting movement in body parts

Body motion	Metric used for detection
Arms	Distance between trackers 2 and 4
Back	Angle between trackers 3 and 4
Legs	Horizontal distance between trackers 1 and 3

When the distance between trackers changes over the duration of the recovery, a range of significant movement can be extracted. To avoid exclusion of users based on their body proportions like arm length, the process of determining significant movement has to be relative to their body. This is done by finding the maximum and minimum distance values and detecting the most significant change in distance between these, based on thresholds. An illustration of this can be seen in Figure 14, where the vertical axis is the distance between trackers 2 and 4, and the horizontal axis spans the recovery phase of a rowing stroke. The lower and upper thresholds, after trial and error, have been set to 23% and

78% of the difference between the maximum and minimum distance respectively. The timestamps of the start and end of the isolated range of movement are the most relevant and are therefore recorded into a matrix.

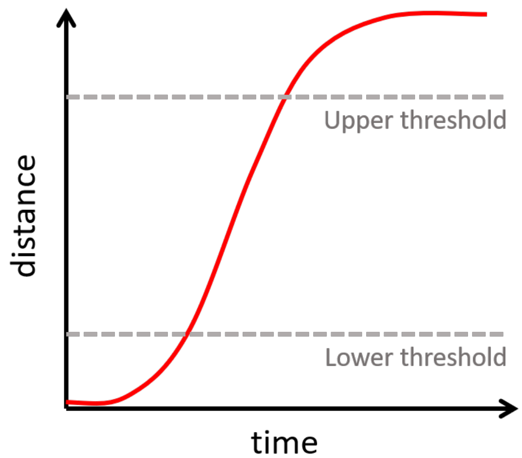

Figure 14: Detection of arm movement over the duration of the recovery phase

While this method suffices, and alternative method for determining this range of motion would be using the derivative, or velocity of the graph. A peak in the derivative would indicate the fastest motion and a range can be extracted around this point. However, in testing, this method proved to output inconsistent results. The reason for this was that moments of tracking inaccuracy led to type I errors, with incorrect peaks being detected. For this reason, the method described above was used instead.

For the back movement, the detection is slightly different. Instead of the distance between trackers, the angle between trackers 3 and 4 are used. This is calculated, in real-time, as follows.

$$Angle_{back} = |\arctan\left(\frac{y_{tracker4} - y_{tracker3}}{z_{tracker4} - z_{tracker3}}\right)| \tag{1}$$

This angle, over time, outputs a similar graph as those of the arms and legs and the process of extracting and recording a range of significant movement is the same as well, using the same threshold values.

5.2.3 Feedback

After the timestamp values of each movement have been recorded, they are remapped between the start and end of the recovery phase of the stroke, as users can row at different stroke frequencies and the values thus need to be relative to the duration of the stroke. The collection of remapped timestamps are then ready to be visualized with the chosen feedback method.

To implement the bar-chart visualization, a few different third-party solutions were tried. After testing, no assets met the requirements of the feedback system, discussed in chapter 4, so the feedback system had to be manually constructed. This was implemented in C# and Unity3D's built-in UI components, and placed in front of the user in the virtual environment, to attract the most attention. Feedback-related code can be found in appendix B.2. The side-view functionality of the feedback system is placed next to it, and consists of a real-time updating texture, rendering the output from an additional camera placed next to the user. Instead of recording the exact virtual environment the user is placed in, the additional camera is rendering selective parts of the world, containing just the user, the trackers and a completely modelled replica of the *RP3 model t* ergometer, invisible to the first-person perspective of the user.

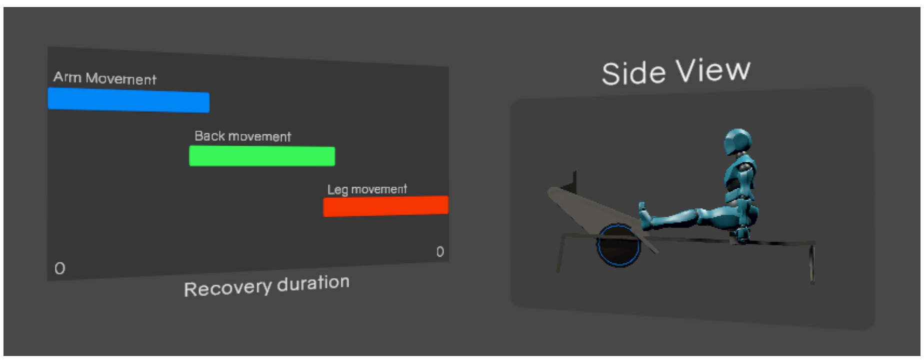

Figure 15: Implementation of feedback systems in the prototype

5.3 Immersion and engagement

After the tracking, and technique analysis, the engagement and entertainment improvements need to be implemented. That includes a humanoid character to resemble the user, a rowing boat moving along a river, and a visually pleasing environment.

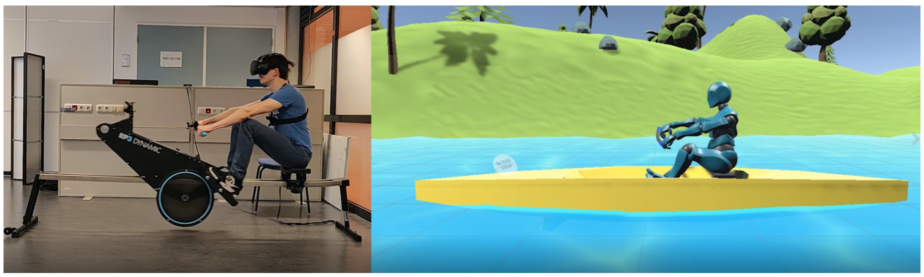

Figure 16: Physical and virtual aspects of the final prototype side by side

5.3.1 Body-matching

A humanoid character is needed to represent the user in the virtual environment. This character model needs to represent a wide variety of users so an amount of generalisation is required. The selection of different characters was chosen to be narrowed down to two different models, one resembling a male person and one resembling a female person, but without any recognisable facial features or skin colour. This distinction avoids racial and other forms of discrimination while still allowing for a level of resemblance. The manual creation of suitable character models would involve character design, modelling, rigging and texturing, which are very time-consuming tasks. For this reason, they were determined to be outside the scope of this project and existing character models from third parties were used, see Figure 17. These characters consist of 3D models, completely rigged and textured.

As specified in section 4.5, the character should also copy the movements of the user as close as possible as to maximise the level of immersion. This, coupled with the limitations of the number of motion trackers, leads to the choice of using inverse kinematics as an animation technique, see section 4.5. To accomplish natural movement, several parts of the skeleton, or rig, of the model were either positioned relative to motion trackers, or animated with the use of inverse kinematics. The overview of joints can be seen in table 3. The positioned joints are noted with the motion tracker it is relatively positioned towards, while the animated joints are noted with the object it is linked to, and the chain of parent joints which are affected. A screenshot from the video[5]showcasing the prototype can be seen in figure 16.

[5]Video showcasing the working prototype, *https://youtu.be/6ccl-kdUBXU*

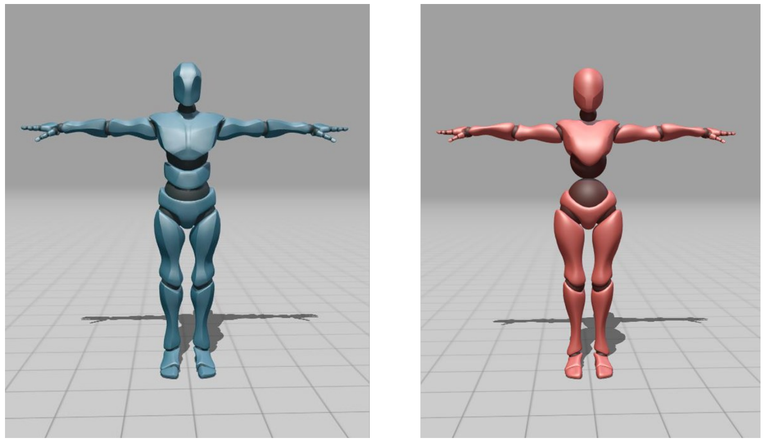

Figure 17: Implemented character models, source: mixamo.com

Table 3: Overview of joint positioning in the character

Joint	Technique	Positioned at
Root	Relative positioning	Tracker 3
Upper spine	Relative positioning	Tracker 4
Right hand	Inverse kinematics	Tracker 2: Ergometer handle, via right elbow, and ends at right shoulder
Left hand	Inverse kinematics	Tracker 2: Ergometer handle, via left elbow, and ends at left shoulder
Right foot	Inverse kinematics	Tracker 1: Right ergometer footrest, via right knee, and ends at root
Left foot	Inverse kinematics	Tracker 1: Left ergometer footrest, via left knee and ends at root
Head	Inverse kinematics	Htc Vive headset, via neck, and ends at upper spine

5.3.2 Water shader

Visualising 3D water in a realistic way is a notoriously difficult thing to implement in games and software. This is one of the reasons the art style of the entire scene was chosen to be more cartoonish. Despite textures and things like foam particles being simplified by this design choice, the movement of water still needs to be accurately visualised in order to create the dynamic feeling of water. Performance, being the impact in frames per second, of the implemented water also has to be considered, since a minimal performance measure of 90 frames per second has to be reached continuously in order to ensure smooth visuals en minimise motion sickness.

On the one hand, regular scripting and animation techniques within Unity, like animations controllers and mono-behaviours, runs its core code on the central processing unit of the computer, which is optimised for large quantities of logic operations. Texturing and shading of objects, on the other hand, runs its core code on the graphical processing unit of the computer, which is far more efficient, but only for these specialised tasks. For this reason, a shader was created for the implemented water, as opposed to scripted and animated water objects. A shader is a highly efficient piece of code that runs on the GPU (graphical processing unit) and manipulates an image or texture with rendering effects before it is drawn on the screen[33]. This efficiency makes for extremely low impact on performance. The tool used to create the shader is the, relatively new as of writing, Unity's node-based *shader graph*. The entire shader implementation can be found in appendix B.4.

For the dynamic effect of water, several systems were implemented. The first is the texture of the water, which is animated by modifying a noise pattern over time. This noise pattern is called Voronoi noise or Worley noise[34] and is based on an algorithm which generates a cell-like pattern, which, when modified, can resemble water, see Figure 18.

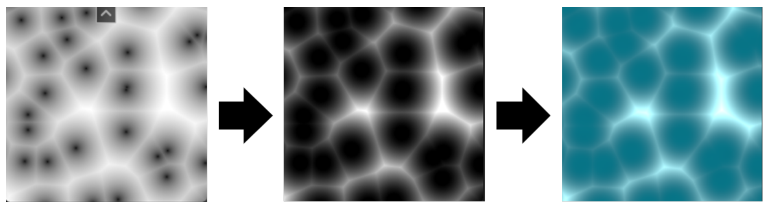

Figure 18: Implementation of voronoi noise in the water shader

Another part of the implemented water shader is the waves. As opposed to just animated textures, this feature actually augments the mesh, or 3D model, of the water object. The vertices, or points, in the flat surface of the water are moved in the vertical axis based on another noise pattern, the gradient noise, see figure 19. This noise is shifted over time and converted to a heightmap, where the pixel colour translates to the height of the vertex.

The last part of the water shader is a foaming effect, which creates a lighter colour where the water intersects with another object, in order to create the feeling of dynamic interaction between the water and environment, as seen in Figure 20. This is done utilising the distance of every pixel, from the camera perspective to objects in the scene, which is called scene depth. Parts of the water texture that are closest to these objects, as seen from the camera, shift in colour.

5.3.3 Rowing boat

Single sculls are a category of rowing boat designed for use by a single rower, using two oars to propel themselves forward[6]. Due to the user capacity of the installation, and the symmetrical nature of the ergometer handle, this is the type of boat chosen to be represented in the virtual environment of the installation, with the goal to increase realism and engagement. The boat was modelled and textured in the program *Maya 2019*, before being implemented in the main scene within *Unity*.

The user should be able to move the boat along the river by using the ergometer. To achieve this, a physics-like movement system was implemented,

[6]https://www.rowpnra.org/pnra/rowing-basics/

Figure 19: Gradient noise used for the wave pattern

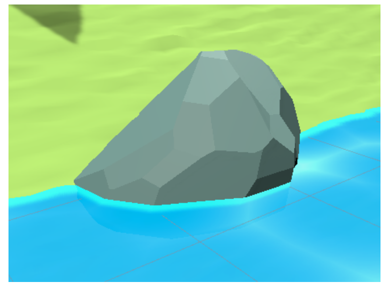

Figure 20: Implementation of foam at the intersection of the water shader with other objects

where the virtual boat has a velocity in the direction of the river. This velocity is visualised on the front of the boat, see Figure 21, and gets augmented with two different forces, applied every physics frame of the simulation (20 times per second). The code for both systems can be seen in appendix B.3.

One of these forces is the acceleration of the boat, caused by rowing. For the most accurate simulation, using the resistance output of the ergometer itself is recommended, but for the scope of this report, the positional data of a tracker on the ergometer handle was used. The velocity of this tracker, e.g. the difference in position between frames, was added to the boat's velocity, after being multiplied with an arbitrary constant, found out by trial and error.

The second force being applied to the boat's velocity is drag, which slowly decreases the speed when not actively rowing. Using advanced mathematical drag model is out of the scope of this research, so after testing a simple linear drag equation, the final implementation was the following equation describing how quadratic drag is applied to the boat's velocity;

$$v_{boat} = v_{boat} + c * v_{boat}^2 \tag{2}$$

5.3.4 Terrain

To construct a visually pleasing environment for users to row in, a representation of a river is surrounded by a hilly terrain consisting of ground, trees, and rocks, see Figure 22. The ground was made in the Unity terrain tool, which allows for generating a height map, which was then customised with hills, mountains, and the riverbed. The terrain was then improved with the addition of tree and rock models, dispersed around the river. Due to the scope of this report, these were retrieved from third parties[7] [8], instead of modelled and textured by hand.

[7]https://assetstore.unity.com/packages/3d/vegetation/trees/free-trees-103208
[8]https://assetstore.unity.com/packages/3d/environments/lowpoly-rocks-137970

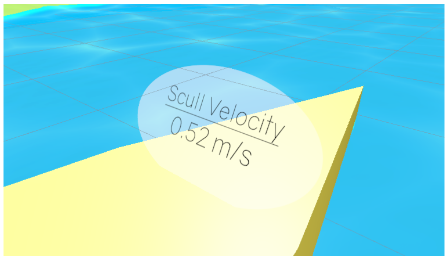

Figure 21: Display of current velocity, mounted on back of the boat

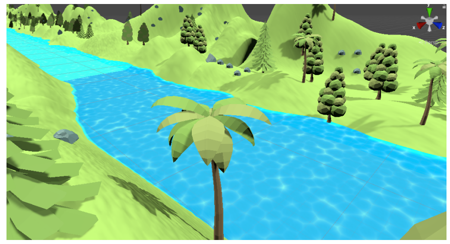

Figure 22: Final look of the terrain and river

5.4 Validation

Before testing with actual users, consultation with experts is required in order to validate the methods used and gain valuable feedback about the implementation. For this, student rowing coaches Abe Winters and Stijn Berendse from rowing association Euros were invited to try out the final prototype. Their comments and feedback were noted down and improvements were made to the prototype after. The results were as follows.

The overall implementation was positively received by the experts, with standouts being the environment, which they described as "Immersive". The techniques used to analyse and view the rowing technique analysis were discussed and approved, although inaccuracies in the tracking were found "Slightly annoying". One additional interesting observation was that they experienced the movement mechanics of the single scull as realistic and very close to the speed of a real rowing boat.

6 Evaluation

As specified in chapter 3, we will conduct user tests to evaluate the installation. In this section, we will explain the setup and goals, and describe the protocol and execution of the user tests, after which the results will be analysed.

6.1 Setup

Conditions

As seen in the research questions of this report, there are a few aspects of the installation that need to be evaluated, namely the effectiveness of the installation's technique correction and immersion level. These will be the dependent variables of our experiment, see table 4 In an ideal situation, the listed variables should be tested individually, e.g. different tests where the new feedback method is changed but using a monitor instead of a heads up display, among other combinations of variables. This allows the specific causation to be better understood, and make sure a possible improvement of technique is actually coming from the motion tracking system and not from the immersive environment for instance.

Independent variables	Dependent variables
Feedback method (Mirror vs bar chart)	Level of technique improvement over time
Environment (Real vs Virtual Reality)	Level of immersion

Table 4: User test variables

The chosen user test, because of the scope of the project, is an all or nothing approach, however, where both variables are measured in a single test. This will follow a between-subjects or between-groups design where subjects are only exposed to one of the conditions. The participants are therefore divided into two groups, a control group is selected and tested without both aspects of the installation enabled, against a group of participants using the new feedback and immersion modifications. Because the subjects are divided into two groups, it is important to have a large sample size. A sample size of 40 is recommended, with a minimum requirement of 20.

Randomisation

Subjects need to be divided into the two groups. According to Suresh [35], a simple randomisation technique used for dividing subjects into groups is performing a coin flip, with both sides of the coin representing one of the two groups. Suresh also states "However, randomisation results could be problematic in relatively small sample size clinical research, resulting in an unequal number of participants among groups."[35, p.9]. This, along with the fact that the expected sample size is on the small side, is why the method was slightly adapted to fit our research. Instead of performing a coin flip for every participant, we performed a coin flip for the first participant of each day of user tests, after which the rest of the participant that day will alternately be put among the two groups. This will presumably keep the selection relatively unbiased, while the ratio of participants will not deviate much from the desired 50/50.

6.2 Procedure

A user test will be conducted where participants will be categorised into two different groups, type A and B, where type A will be the control group, measured using a traditional ergometer setup and type B will be measured while using the installation. The user test protocol can be divided into the following steps:

- Explanation
- Informed consent
- Pre-experiment survey
- Instruction
- Null measurement
- Practice period
- Final measurement
- Task Evaluation Questionnaire (IMI)
- Interview
- Debriefing

Explanation

Subjects to experiments are owed a short briefing before participating in an experiment. This will be done during the recruitment of participants as well as in this short briefing and contains a general overview of the experiment procedure as well as a short description of the research, without telling them the conditions of the experiment.

Informed consent

If any monitoring or measuring is to take place, participants will, of course, have to fill in an informed consent form, as in good practice in research and per GDPR regulations. The types and amount of data collected and possible risks of the experiment are explained in a short description, which is different for participants of type A and B because the data collection and risks are different. An example of this is the (low) risk of motion sickness for type B participants who will be wearing a VR headset. The two consent forms can be found in appendix C.1.

Pre-experiment survey

Before instructing participants to use the installation, they asked to fill in a small survey, which can be found in appendix C.5. This is done for two reasons. First, to gather some basic information like age, height and gender, used for the analysis of the results, and second, to screen participants for attributes which would exclude them from the research. This involves asking the participant about their rowing experience. Participants with rowing experience exceeding a few lessons have most likely already mastered the technique improvements offered by the system, and are therefore excluded from the research.

Instruction

For the participants to use the ergometer setup, with or without Headset, a small instruction is required. Participants, of both types, are told the exact same instruction, validated by a student rowing coach, along with a video example of proper technique, recorded by a student rowing coach, see appendix C.4. The technique explained is limited to the scope of this project's technique correction, being timing and overlap of the different phases in a rowing stroke. Then, participants are explained the exact procedure of the user test, as described in this section.

Null measurement

A null measurement is conducted where a baseline rowing technique is established for each participant, regardless of type. They are asked to row 10 strokes on the ergometer in order to get an accurate result. This is done with the full tracking system, including chest tracker to output a technique report, but without VR headset. This null measurement provides a comparison point of technique level which we can later use to draw a line of (possible) improvement.

Practice period

To measure the level of improvement in rowing technique, participants need time to practice their technique either with the installation or with traditional methods. Type A participants are provided with a large standing mirror next to the ergometer to get feedback on their technique, as seen in figure 23, while type B participants are provided with the VR headset and virtual feedback system. While it is desired to let participants have as much time as possible to practice, due to the limited available time, a practice period of 5 minutes is chosen. This allows participants to acquaint themselves with the installation and have time to improve on their technique while not exhausting them too much.

Final measurement

After the practice period is complete, the VR headset or mirror is removed and the participants are asked to row 10 last rowing strokes while wearing

Figure 23: Standing mirror next to the ergometer for user tests of type A

the tracking system. This technique data can then be compared to the null measurement, so a difference in technique level can be determined.

Task Evaluation Questionnaire

The Intrinsic Motivation Inventory (IMI), grounded from the Self Determination Theory [36], is a method of measuring the subjective experience of participants in an experiment. It is a multi-item questionnaire that gives a set of guidelines for assessing subjects' interest/enjoyment, perceived competence, effort, value/usefulness, felt pressure and tension, relatedness and perceived choice during an experiment. These subscales contain a set of questions on a likert scale that have been shown to be analytically coherent within their subscale, across a variety of conditions, meaning that if a user were to answer 5 on a question, they should give a similar answer to different questions within the same subscale. If results show mixed answers within a subscale, resulting in a low cronbach's alpha score, the reliability of the category might be unreliable, as the subject might not fully understand the questions. As stated by the IMI, the given questions can be slightly modified to fit specific research or experiment.

When applying the IMI to this report's user tests, a slightly different version of the base questionnaire is used. This version is called the *Task evaluation questionnaire*, and has been proven to be reliable as well. This questionnaire excludes subscales like relatedness as they are not relevant or applicable. The final question list can be found in table 34, the (R) in the second column denotes that the question should be inversely scored, as it measures the opposite of the given subscale.

Interview

The interview after the experiment is done for three reasons. First, to ensure the results from the measurements are accurate and fair, by confirming participants were using the installation in an intended way. For instance, if they are actually using the provided tools in order to improve on their technique, like the mirror or the feedback system. The second reason is to get feedback and quotes from participants by asking some open questions about their experience. This can provide us with valuable observations and effects of the installation that were not expected. The last reason is to double-check for adjusted behaviour by asking if they had any knowledge about the different conditions of the experiment.

Debriefing

At last, participants are fully disclosed all information and conditions of the experiment, and what it will be used for. They will be rewarded with a small compensation in the form of a candy bar.

6.3 Execution

6.3.1 Pilot test

After defining a protocol, a short pilot test was conducted. This pilot test has a small sample size of 1 (type B) and serves to improve the protocol and prototype if necessary. A participant was selected using convenience sampling, and the entire user test was conducted, without analyzing the results. After the pilot test was completed, a few observations were made.

Concerning the prototype, a few aspects were found to be not sufficiently working and thus needed to be improved before conducting the full user test. One of these was the stroke detection, which consistently failed during the pilot test. The system detected double stroke as well as no strokes at all sometimes. While tests during the realization phase were successful, it appears that the accuracy of stroke detection heavily depends on the height and proportion of the user, as the pilot participant was significantly longer than I am and had shorter arms relative to their height. Another aspect was that the back of the virtual character was sometimes obscuring the first-person camera perspective in the virtual environment. This is due to the participant leaning very far backwards, which the virtual character was not able to do.

To modify the prototype before the user tests the following was done; First, a completely new stroke detection algorithm was ideated and tested, based on real-time peak detection. When this was not able to be finished in time, the method was rejected. Instead, there was chosen for an augmented version of the existing detection algorithm, with tweaked parameters. After a few tests with both me and the same participant, the detection was deemed adequate. The back problem was only partially able to be fixed. The inverse kinematics method of the rotating back was changed by shifting the joint from which the upper back rotates one down. This is not a perfect solution, however. An alternative method could be the customisation of the user's character, which also would enable the user to relate more to their virtual avatar, on aspects like height, proportions, gender and skin colour. This method was left out due to the scope of this research.

6.3.2 Main user test

For the actual user test, invitations were sent out to students and faculty members at the University of Twente. This convenience sampling method leads to a sample size of 20 participants, which is the bare minimum of the requirements set out at the start of the chapter.

During the user test, a significant hurdle occurred. Due to last-minute improvements to the logging system in the prototype, a bug prevented the logging system from functioning at all. This was exposed after the first three participants had completed the experiment, which means until that point, no actual motion data was recorded. After repair of the logging system, the user tests could continue with the exclusion of the first three subjects, and extra participants were recruited to reach the sample size of 20.

6.4 Results

On completion of the user test, a large amount of data is collected. For every participant, this data consists of the following: The filled in answers on both pre- and post-experiment surveys/questionnaires, the dictated answers to the interview, the informed consent form and many unity-produced logs, containing information on rowing technique, both from before and after the 5-minute practice period. This section will set out to analyze and visualize this data.

6.4.1 Data structuring

To visualize results and draw conclusions, this information first has to be organized. This subsection will describe the process of importing, structuring and filtering data from the user test.

Questionnaire

Before the results of the IMI questionnaire can be statistically analysed, the questions need to be converted to the given scales, being enjoyment, perceived competence, perceived choice and pressure/tension. The first step is to reverse score the inverse questions by substracting the answered number, on a scale from 1 to 5, from 6. After doing this the questions are grouped per category/subscale, as given by the IMI [36]. Per participant, all questions within subscales are then averaged and converted to percentages. The results of this can be seen in appendix D.1.

Technique

Then, the unity-produced logs need to be analyzed. This is done by importing the logs, per participant, in the program Matlab and parsing the text based on headers, see appendix D.1. Once the movement timings are extracted, the logs are separated, because the strokes before and after the 5-minute practice period need to be compared to each other. Because the logs themselves do not contain information about the type of measurement, this is done based on timestamp. A time gap of larger than 4 minutes between logs is detected as the practice period. Then it is time to associate scores with every stroke/log. The method for this was verified by student rowing coaches, see equation 3, and goes as

$$\begin{aligned} gaps_1 &= (Start_{back} - End_{arms})/\Delta t \\ gaps_2 &= (Start_{legs} - End_{back})/\Delta t \\ gaps_3 &= (Start_{legs} - End_{arms})/\Delta t \\ Score &= \frac{1}{3}\sum_{i=1}^{3} gaps_i \end{aligned} \tag{3}$$

where *start* and *end* indicate the timestamps where body parts start and stop moving respectively, and Δt the duration of the stroke. When the scores are calculated for every stroke, all logs from before the practice period are averaged, as well as the logs after. The difference in these two means can be seen as the improvement over the practice period, see equation 4.

$$Improvement = \frac{1}{n_{after}}\sum_{i=1}^{n_{after}} after_i - \frac{1}{n_{before}}\sum_{i=1}^{n_{before}} before_i \tag{4}$$

Before describes the collection of scores from before the practice period and *after* the scores from after the period. With the improvement in technique determined for every participant, statistical analysis can be done.

6.4.2 Questionnaire analysis

Processed results for every category of the task evaluation questionnaire can be seen in figure 24, which depicts the values for every category for the two groups. This section sets out to analyze the processed data by comparing the two groups A and B. Methods will be described and results from the analysis will be given.

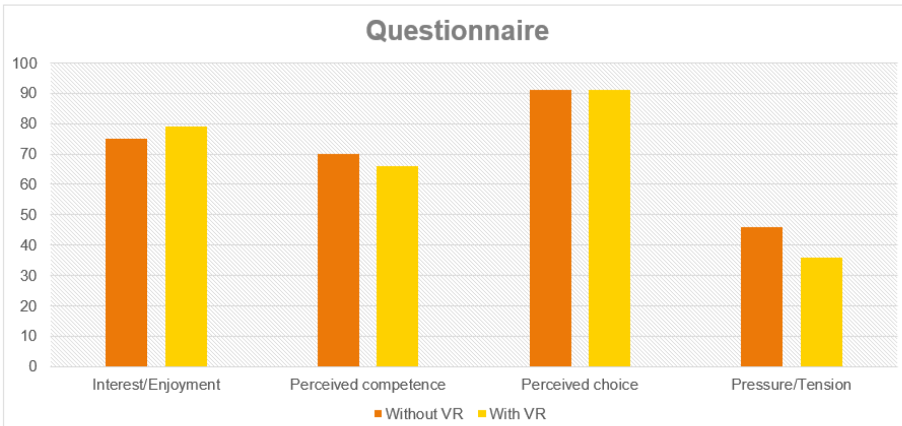

Figure 24: Visualized averages of questionnaire subscales, per group

Reliability analysis

To check how conclusive the results are, a reliability test needs to be done. For this, Cronbach alpha scores are calculated for every subscale of the task evaluation questionnaire, in the program SPSS. The results of this can be seen in table 5. A Cronbach's alpha score indicates how aligned a set of answers are on a scale of 0 to 1 (e.g. a score of 1 means that all values in the set are the same). Applied to the questionnaire subscales, this tells us something about how consistent a participant answers questions within the subscale, and thus the degree of reliability of the category.

Subscale	Cronbach's alpha score
Interest/enjoyment	0.896
Perceived competence	0.824
Perceived choice	0.751
Pressure/tension	0.300

Table 5: Reliability analysis of questionnaire subscales

6.4.3 Interview questions

The open questions from the interview resulted into a few interesting results. Besides checking if participants knew the complete experiment with conditions, which none of them knew, the open questions provide us insights into the thought process of the subjects. In table 6, an overview can be seen of common remarks made by the participants, sorted by condition.

Table 6: Overview of remarks during the interview

Remark	Frequency (Control)	Frequency(VR)
I actively used the provided feedback systems	80%	90%
I found the provided feedback systems to be complicated	0%	30%
I experienced motion sickness	0%	10%
I would do this again	0%	20%

6.4.4 Technique analysis

Normality test

Before any comparison can be done to analyse differences in technique improvement, the distribution of samples has to be determined first. This is because different types of distributions have different methods of statistical comparison. The hypothesis is that the samples come from a normally distributed population, as there are no anomalies that would indicate otherwise. The first step is therefore to apply a Shapiro-Wilk's test of normality. The result of this test can be seen in appendix E.2.1 and is interpreted as follows.

The Shapiro-Wilk test outputs a p value between 0 and 1. Because the null hypothesis is that the samples come from a normally distributed population, normality is assumed for p values higher than the chosen alpha level. If the p value is lower than the alpha level, the hypothesis is rejected and the population is assumed to be not normally distributed. The alpha level indicates a level of confidence and, for the purposes of this report, is set to 0.05. This means that we can be 95% confident about the outcome of the test.

As seen from the results, the p value, or significance value, is 0.500 for the group using a mirror, and 0.749 for the group using VR. With the alpha level of 0.05, the null hypothesis is accepted for both groups, meaning that all of the samples are assumed to come from a normally distributed population.

Comparison

For the final comparison between the two groups, an independent-samples T-test will be done. This statistical technique gives an indication of whether two collections of samples have significantly different means from each other. The null hypothesis is that the two groups are equal in mean, with the alternative hypothesis of different means. When given the technique improvement scores for every participant, per group type, SPSS will output both a t value as well as a p value. The first of which can be compared to a t-table. With a confidence level of 95% and a sample size of 20, the p value can be determined to be 0.13, which is higher than the alpha level of 0.05. This means that we accept the null hypothesis and the two groups are determined to come from the same population.

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
improvement_score	Equal variances assumed	2,499	,131	,259	18	,799	3,50900	13,54982	-24,95811	31,97611
	Equal variances not assumed			,259	14,799	,799	3,50900	13,54982	-25,40598	32,42398

Figure 25: Results from an two sample independent T-test

7 Conclusion

The conclusion of this reports will start by summarising the findings from the evaluation, after which conclusions can be drawn about the effectiveness of the prototype and the opportunities for VR application in indoor rowing. The section will end with answering the research questions posed at the start of this report.

Findings

From the results of the user test, a few findings can be extracted about the possible immersion and technique improvements. The first thing is that the enjoyment of using the installation goes up slightly for participants using VR, while perceived competence goes down slightly. An interesting but more unexpected outcome was that pressure/tension went down considerably, although with a low reliability score. Furthermore, the interview with open questions showed that participants found the VR solution more enjoyable, and would repeat the experience willingly more often than participants using the traditional method. On top of that, the small expert validation showed promise, as experts stated the boat movement seemed very realistic to the real rowing experience, as well as the VE being more engaging to row in than rowing on the ergometer alone.

Regarding the technique improvement, the user tests did not show any significant difference between traditional methods and the proposed method on technique improvement using VR. A reason for this could be the relatively low sample size of 20 participants, or inadequacies in the prototype, as some participants stated that the feedback system was complex and the tracking inaccurate. However, the results still point to benefits on technique improvement. The reason for this is that, although there seems to be no difference between groups, the mean improvement in technique over the practice period is still more than 20%. This can be, at least party, contributed to the VR solution, as almost all participants stated actively using the two provided feedback methods in the VE, in order to improve on their rowing technique.

Answering research questions

With these findings, concrete answers can be given to the research questions of this report, starting with the sub-questions.

- Can rowing technique be analysed and corrected using VR motion trackers and digital feedback?

Yes, it seems so, although not better than traditional methods. More research is required to find more conclusive results.

- Can the immersion of indoor rowing be improved using a head-mounted display?

This question can be answered with a cautious *yes*, as the results show slight immersion and tension improvements.

- How can Virtual Reality improve the current state of indoor rowing?

This research shows that Virtual reality technology, namely motion tracking and an HMD with an immersive environment, positively contribute to the state of indoor rowing in two aspects; technique analysis and immersion, as seen in observations and from user tests.

8 Discussion

This section will go over the quality of this research by discussing the execution of the conducted research, prototype and evaluation. After this, recommendations can be given about future work on this topic.

Prototype

As stated in the conclusion, the built prototype has some issues that either fell out of scope of this research to fix, or came to light during the evaluation phase. One of these is the technique analysis. This might come from the small number of trackers, only four, the velocity of the tracking devices in a small space, or inadequate algorithms. Another aspect of this issue is the hardware used. The tracking devices are promising but the accuracy of technique correction can, presumably, be enhanced significantly when accessing sensor output from the ergometer itself, as well as motion tracking data. This produced force curve is already heavily used in traditional technique analysis and can be a valuable addition to the installation.

A second issue with the prototype is to do with the character. The current humanoid character, controlled by inverse kinematics, has some shortcomings found during the evaluation phase. That is because a single character model cannot resemble every possible user in the installation. Usage of the installation by people with bodily proportions which mismatch the character model, such as longer arms, will result in incorrect visualisation of the user which might impact the immersion.

The last problem with the realisation of the prototype is with the feedback system. As users reported that they did not understand the current implementation, improvements have to be made. A reason for this complex interpretation might be a high cognitive load on the user, as discussed in chapter 4.

Evaluation

While the results from the evaluation seem to fit the observations and interviews, a few observations can be made about aspects of the execution that can be improved. The first and most important issue is the sample size of 20. This is on the low side to draw conclusive results from and is presumably one of the reasons the difference in groups is insignificant. A sample size of 40 or above is recommended.

The outcome of the questionnaire is an interesting point of discussion. While the slight increase in enjoyment is expected, the decrease in pressure tension was not. Reasons for this might be increased immersion, relaxation of rowing in a visually pleasing environment, or the fact that an HMD makes the user unaware of the researcher observing them during the user tests.

Another important aspect is the setup of the experiment itself. The current setup suffices to show visible advantages of the proposed solution, however clear causation is still unknown. This is because multiple dependent variables were tested against multiple independent ones (e.g. both level of immersion and technique improvement). While it is expected that the causes of any difference in technique improvement between the groups are caused by the technique analysis and feedback, we cannot be sure since the immersion factors like a detailed environment are also added into the mix.

The last point of discussion, is the participant variation in the user tests. The method used for inviting participants was convenience sampling, which lead to some unwanted results. Almost all participants appeared to be students who were studying the *Creative Technology* bachelor. This might be an issue as it is not uncommon for Creative Technology students to be accustomed to VR technology or even know some information about this research. While subjects with significant VR experience are filtered out using the survey, a more diverse user group is still preferable.

8.1 Recommendations

Apart from small things noted in the discussion, there are some recommendations to be made which can be researched in future work on the topic. The first and most important recommendation is to further specify future research into a few directions like feedback, tracking and technique, or immersion. This is because of the broad nature of this research, showing a proof of concept, as well as the unclear causation seen in the results from this report. Directions recommended for future research go as follows.

The first direction would be the design of a feedback system for rowing technique analysis, with a low cognitive load. This would entail evaluating conceptual designs for a feedback system based on their complexity and usefulness. In addition to traditional UI elements, audio visual cues could be used as methods, like rhythmic sounds to encourage consistency. Designing many different options for this aspect of the prototype fell out of scope for this report.

Another recommended direction for future work is research into methods for analysing rowing technique. This is because the accuracy of tracking in the current prototype is not as high as hoped for. On top of trying different hardware systems, methods have to be found or designed for evaluating tracking accuracy in rowing technique analysis, as that has not been done academically in this report. In addition to VR motion tracking devices, alternative sensing methods could be used for the implementation, such as the force curve output of a connected ergometer.

The immersion benefits of VR in the context of indoor rowing is also recommended for further research. The implementation of the prototype in this report consists of many different aspects which aim to improve immersion, so research into the effects of isolated methods could be done. The aspect that shows the most promise is using body-matching techniques to improve immersion. Future research into the effects of body matching specifically, in the context of indoor rowing, could be very valuable. Problems like the character resemblance inaccuracy could be solved when trying more advanced body-matching implementations, like adjustable character models or different *Inverse kinematics* solutions.

9 Ethics

In this section, ethical implication of the research will be reflected upon. This is done by analysing and applying an ethical toolkit for engineering and design practice[9]. This toolkit consists of seven steps which can be executed for ethical analysis on an engineering or design project.[10]

9.1 Ethical risk sweeping

In this section, the first step of the ethics toolkit will be applied on the graduation project, thereby analysing possible ethical risks and answering the question; Which design choices can possibly lead to harm or ethical negligence?

Injuries

Injuries in rowing is very common, in boats as well as on the ergometer. While the goal of the graduation project is to improve on the current state of this, there is always a possibility of users injuring themselves while using the installation. An important thing to consider is therefore 'who is liable for injuries?'. To cover for this, a collection of warnings and evaluated user guides need to be provided for the users in order to ensure correct usage and cover liability. On top of this, while injuries are well covered and analysed on regular ergometers, the installation for this graduation project works significantly different. Mainly because instead of seeing their body in physical activity, users wear a headset which provides them with a different and inaccurate representation of their body in the virtual environment. As the designer, I have to account for this difference and make there is a low chance of injuries.

Unintended uses

While the project is made with the intention of improving on an individual user's technique, the installation could technically be used by organisations to evaluate rowers. This poses a big risk to the project, because it cannot guarantee that it evaluates correctly. One of the reasons for this is that ergometer training differs significantly from actual rowing on the water so ergometer performance should not be used as a benchmark. Another is technique bias, see section 1.5. Altogether, because of unintended uses, there is a great responsibility to make accurate and evaluated corrections for the user.

Motion sickness

A known problem of Virtual Reality in general is motion sickness, or cybersickness [19]. This can have several causes but the most important one is the dissociation of what users feel and what they see, this is even more prevalent in situations where physical activity or rapid movements are involved. There-

[9] Retrieved from *https://www.scu.edu/ethics-in-technology-practice/ethical-toolkit/*

[10] This section is partly extracted from my own Reflection II final essay assignment, as part of the Creative Technology module 12 curriculum.

fore the Virtual Rowing installation has a very real chance of inducing motion sickness in its users.

Sensitive data

To analyse the rowing technique, a selection of sensors is used to gather data from the user. This data might contain privacy sensitive information in the future, because personal information like height, sex etc might be linked to the user's performance and technique analysis. Therefore a few aspects need to be considered; Will the data be stored? If so, online or offline? will the data be anonymised? Will it be encrypted? And finally, will the data be shared? What are the consent options?

Technique bias

Despite a universally agreed upon base technique, there are still many discussions among experts about what the perfect rowing technique looks like. If the project is to correct the technique in a positive manner, the baseline used to compare the users technique with is very important. The responsibility lies with me to find a compromise among different discussed techniques and make sure a an objective correction is made that works for a diverse group of users. If not accounted for, the risk is that users will learn and get used to a certain form of rowing technique which might not be right for them.

Feedback system

The appropriate manner of feedback needs to be considered. Due to the time between strokes of the rower being very short and the actual amount of data generated on rowing technique per stroke is large, there is not much time for the user to interpret feedback, so decisions have to be made of what is and what is not visualised. This could contain biases because people have different views on what the most important data is to be shown.

Privacy

With a head mounted display attached, the user is completely unaware of their surroundings. On top of that, the project will provide a secondary display for outside viewers, to get a top down view of what is going on in VR, this is necessary for coaches to inspect the training. This is a potential privacy problem though, because the user has no awareness or control over who sees this, while there is (potentially sensitive) information provided on said screen about the quality of the user's technique.

9.2 Ethical pre-mortems and post-mortems

While this tool intends for both pre- and post-mortems to be analysed, this section will only go over pre-mortems, due to the incomplete nature of the graduation project making it impossible to reflect on problems that have already occurred. The section will consist of a few thought out scenarios where a systemic failure leads to an ethical disaster.

Rowing team selection

Imagine a high profile rowing team, like a national Olympic team, looking to expand their selection of athletes with new members. Instead of a regular evaluation of rowing performance, like a sprint or endurance rowing session on the water, the management chooses to use the Virtual Rowing project for selection. Every applicant or invited prospect will go through a process in which they use the installation to their maximum efforts, after which the generated results, which contain rowing performance and technique information, will be compared.

The management then uses this comparison to select a rower with the most desirable results, to be added to the team. Suddenly the implications of the Virtual Rowing evaluation system are far more serious and can have a large negative impact on the lives of its users.

Security Breach

In this hypothetical scenario, the Virtual Rowing installation is relatively widely used and thus gathers a lot of data, which is all personalized to a user's height, sex and in an ethically risky situation, also to a user's name or account. While this on its own, might not lead to many problems, let's suppose all this data is stored in an online database instead of locally, in an unencrypted manner. A foreign 'hacker' with ill intentions could succeed in breaching the server's security and stealing the personal information of thousands of users. Because of the fairly sensitive nature of this data, when published or sold, it could damage the reputation of rowers. These security issues do not cause a problem on their own but altogether they cascade into a bigger issue, consulting with cyber security experts is therefore recommended.

High profile injury

In the last scenario, a publicly well known rowing athlete uses the Virtual Rowing setup to evaluate their rowing technique. Because of their high skill level, the results are very positive. This specific athlete had not had any experience with Virtual Reality however. Due to this, the effects of motion sickness and sensory dissociation [15] have a greater and in a moment of physical discomfort, the user injures their leg. An injury on its own can happen, but because of the high profile aspect of the specific athlete, not being able to row professionally for months after the incident, can be a large issue and will probably lead to a large monetary damage as well.

9.3 Expanding the ethical circle

When working individually or in a small team, it is very common to oversee some problems or risks, because the scope of harm is not well anticipated. Therefore, it is key to understand all stakeholders. This section will go over several types of stakeholders, how they will be affected and what the related risks are.

There are a few stakeholders to consider in the Virtual Rowing project. The first and most obvious one would be entry level rowing athletes themselves. They are the most affected by using the installation, as discussed in previous sections. It is important to inform these stakeholders before use of all possible risks so they can fully consent and thereby accept possible consequences. While requirements state that a low skill level of rowing and some experience with Virtual Reality is needed, users that do need meet these criteria should be taken into account.

A second, unintended group of stakeholders might be casual gym-going people who try it as an alternative to regular ergometer training. These people are not looking for feedback on their technique and have a lower chance of having Virtual Reality experience.

Another possible group of stakeholders are rowing coaches. While Virtual Rowing is not meant to displace these professionals, there is a lot of overlap in functionality so there is a real possibility of actual coaches being affected by wide use of this project, more insight is required, for instance by involving coaches in the design process. Finally, rowing club or team management need to be considered. They would not only be the ones causing widespread adoption and implementation, but there is a possibility of significant structural change inside these organisations. Both intended and unintended, these implications need to be analysed; can the project be used for other purposes than technique correction? When widely used, how does Virtual Rowing impact these stakeholders?

9.4 Case-based analysis

This section will go over some researched cases from the past, relevant to this solution, and reflect on what the reasons were of the ethical problems that arose. Then parallels to the Virtual Rowing project will be drawn in order to find ethical lessons to be learned.

<u>Data breach</u>

Yahoo! Is a large tech company owning several web services like email, calendar and forums. In 2013 one of the largest data breaches ever occurred, where 3 billion records were stolen from Yahoo by unknown cyber criminals. The reason was ultimately poor security on the company's part. Criminals first got access to Yahoo's internal network by a phishing link set via mail to one of their employees. Once compromised, they entered the company's database and stole personal information of millions of users like contact details, calendars and email content. Victims are still able to claim part of the settlement lawsuit.

There are several problems on Yahoo's part that together caused this massive security problem. The first of which is poorly designed database security, which is easily compromised, presumably without encryption. The second problem is access control; in a well structured company, access to sensitive information is restricted and in no situation should every low level employee have access to the central database.

When comparing the case to the subject of this report, a few parallels can be drawn. While access control is an issue only relevant when expanding the solution into a larger company, database security can already have severe implications. Sensitive data storage has to be considered when reflecting on this case.

College admissions

In U.S colleges and universities, it is common to obtain a scholarship if your performance in football or other college sports is very good, so you could represent their college or university in their team. Because these scholarships can reach up to enormous sums, sports performance can be crucial for many high school athletes. In this case, a californian man was convicted for falsifying a huge amount of evidence of sports competency of high school students, commissioned by their parents. Many of these students received the scholarships and got into college.

The most important parallel to be drawn from this case is about the evidence. The results from the Virtual Rowing project are, of course, a similar proof of competency, and as seen from this case, have therefore the possibility to impact the lives of many people. Not only for colleges and universities but (national) rowing teams as well. Regardless of the intent, designers should take great responsibility in designing feedback of the results.

9.5 Remembering the ethical benefits of creative work

When analysing the ethical implications of your work, it's easy to only think about the risks and dangers, and forget that the results of your work can actually do ethical good in the world. This section will go over a few ethically positive impacts Virtual Rowing can have on the world.

Less injuries

In the most ideal case, the Virtual Rowing solution is widely adopted and installed on many rowing clubs and teams. Newly added and low skill level members would be using the installation actively to improve on their rowing technique, meaning that fewer coaches are needed. Hopefully, their technique improves drastically, which means that the chance of injuries is drastically reduced. This is especially good if viewed from a utilitarian perspective, since the health of many people would improve.

More rowers

The aforementioned decrease of coaches might sound bad, but the efficiency increase has positive implications as well. First, the cost of rowing training might go down significantly, meaning the barrier of entry goes down and more people might be tempted to start rowing. This is even more enforced by the next point.

Higher engagement

While the main focus of the project is on technique improvement, there are still a visually pleasing environment and some performance stimulating features to be found, which improves on the engagement of its users. This entertainment aspect of the Virtual Rowing project could not only entice more users to start rowing, but also increase the enjoyment significantly.

9.6 Think about the terrible people

After analyzing ethically positive impacts the project can have, it is also important to prepare for the worst, because there are always people in this world who will use your design with ill intentions. This section will therefore find all intended and unintended ways the current solution can be used, what the opportunities or incentives are for mis-use, and how we might fix them.

One of the unintended uses of this graduation project, as stated in section 4, is using the evaluation results from the error detection as criteria for admissions. This could be college admission, olympic team admission, gym entrance etc. While not failsafe, one of the ways to discourage abuse of this type, is to construct the feedback in such a way that it is impossible to extract or export the results, with the goal to make it harder for the 'terrible people' to gain access. Another way might be to limit the amount of feedback. For instance not showing detailed information about rowing stroke but instead only show short sentences like "Try to bend your back more". This might decrease the usefulness of the results for people who might mis-use this. Just make sure to not lower the effectiveness of the technique correction.

Another example of terrible people is hackers. If ill-minded cyber criminal were to gain access to the Virtual Rowing Database and code, there could be a lot of harm done. Not only could they steal sensitive personal data, which could be leaked, sold or blackmailed, they could also compromise the core functionality of the graduation project. Once tampered with, this could lead to the system giving inaccurate results, thereby teaching users flawed technique and indirectly increasing the chance of injuries.

To combat hacking, a cyber security team or expert needs to be involved. They would have the responsibility of protecting the system against cyber attacks and ensure code integrity.

9.7 Closing the loop, ethical feedback and iteration

One of the most important factors in making sure your product is succeeds ethically, is realizing that reflecting on ethical implications is not a one-time thing. Because norms and values change over time, the process should be an on-going one. Iterating on ethical values of your product or company in a consistent manner to make sure it stays a moral success for a long time.

In the initial design stage this might look like considering what the ethical implications might be, for each new step or addition to the product. For later stages though, it is a matter of planning regular meetings with a dedicated ethics team, or in the case of this graduation project, with every involved designer, to go over changes in moral values and reflect on the ethical ways the project has been impacting society.

References

[1] F. Wilson, C. Gissane, and A. McGregor, "Ergometer training volume and previous injury predict back pain in rowing; strategies for injury prevention and rehabilitation.," *British Journal of Sports Medicine*, vol. 48, pp. 1534–1537, nov 2014.

[2] T. Smoljanovic, I. Bohacek, J. A. Hannafin, O. Terborg, D. Hren, M. Pecina, and I. Bojanic, "Acute and chronic injuries among senior international rowers: a cross-sectional study.," *International Orthopaedics*, vol. 39, pp. 1623–1630, aug 2015.

[3] E. McNally, D. Wilson, and S. Seiler, "Rowing injuries.," *Seminars in musculoskeletal radiology*, vol. 9, pp. 379–396, dec 2005.

[4] D. L. Neumann, R. L. Moffitt, P. R. Thomas, K. Loveday, D. P. Watling, C. L. Lombard, S. Antonova, and M. A. Tremeer, "A systematic review of the application of interactive virtual reality to sport," *Virtual Reality*, vol. 22, no. 3, p. 183–198, 2017.

[5] W. Ijsselsteijn, Y. D. Kort, J. Westerink, M. D. Jager, and R. Bonants, "Fun and sports: Enhancing the home fitness experience," *Entertainment Computing – ICEC 2004 Lecture Notes in Computer Science*, p. 46–56, 2004.

[6] S. Deterding, R. Khaled, L. Nacke, and D. Dixon, "Gamification: Toward a definition," in *CHI 2011 Gamification Workshop Proceedings*, (Vancouver, BC, Canada), 2011.

[7] P. Buckley, S. Noonan, C. Geary, T. Mackessy, and E. Nagle, "An empirical study of gamification frameworks," *Journal of Organizational and End User Computing*, vol. 31, pp. 22–38, jan 2019.

[8] E. Tuveri, L. Macis, F. Sorrentino, L. D. Spano, and R. Scateni, "Fitmersive games: Fitness gamification through immersive VR," in *Proceedings of the International Working Conference on Advanced Visual Interfaces - AVI '16* (P. Buono, R. Lanzilotti, and M. Matera, eds.), (New York, New York, USA), pp. 212–215, ACM Press, jun 2016.

[9] S. Yoo, C. Ackad, T. Heywood, and J. Kay, "Evaluating the actual and perceived exertion provided by virtual reality games," in *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '17*, (New York, New York, USA), pp. 3050–3057, ACM Press, may 2017.

[10] E. Luckett, T. Key, N. Newsome, and J. A. Jones, "Metrics for the evaluation of tracking systems for virtual environments," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 1711–1716, IEEE, mar 2019.

[11] F. Hülsmann, J. P. Göpfert, B. Hammer, S. Kopp, and M. Botsch, "Classification of motor errors to provide real-time feedback for sports coaching in virtual reality —a case study in squats and tai chi pushes," *Computers & graphics*, vol. 76, pp. 47–59, nov 2018.

[12] J. M. Wilken and B. J. Darter, "Virtual reality and real-time feedback to improve gait performance in a polytrauma patient," in *2008 Virtual Rehabilitation*, pp. 71–71, IEEE, aug 2008.

[13] D. L. Eaves, G. Breslin, and P. van Schaik, "The short-term effects of real-time virtual reality feedback on motor learning in dance," *Presence: Teleoperators and Virtual Environments*, vol. 20, pp. 62–77, feb 2011.

[14] J. Varela-Aldás, E. M. Fuentes, G. Palacios-Navarro, and I. García-Magariño, "A comparison of heart rate in normal physical activity vs. immersive virtual reality exergames," in *Human systems engineering and design II* (T. Ahram, W. Karwowski, S. Pickl, and R. Taiar, eds.), vol. 1026 of *Advances in intelligent systems and computing*, pp. 684–689, Cham: Springer International Publishing, 2020.

[15] R. L. Kohl, "Sensory conflict theory of space motion sickness: an anatomical location for the neuroconflict.," *Aviation, Space, and Environmental Medicine*, vol. 54, pp. 464–465, may 1983.

[16] G. E. Riccio and T. A. Stoffregen, "An ecological theory of motion sickness and postural instability," *Ecological Psychology*, vol. 3, pp. 195–240, sep 1991.

[17] J.-R. Chardonnet, M. A. Mirzaei, and F. Mérienne, "Features of the postural sway signal as indicators to estimate and predict visually induced motion sickness in virtual reality," *International Journal of Human–Computer Interaction*, vol. 33, pp. 1–15, jan 2017.

[18] U. A. Chattha and M. A. Shah, "Survey on causes of motion sickness in virtual reality," in *2018 24th International Conference on Automation and Computing (ICAC)*, pp. 1–5, IEEE, sep 2018.

[19] C. Regan, "An investigation into nausea and other side-effects of head-coupled immersive virtual reality," *Virtual reality*, vol. 1, pp. 17–31, jun 1995.

[20] S. F. M. Zaidi and T. Male, "Experimenting novel virtual-reality immersion strategy to alleviate cybersickness," in *Experimenting novel virtual-reality immersion strategy to alleviate cybersickness* (S. N. Spencer, ed.), (New York, New York, USA), pp. 1–2, ACM Press, nov 2018.

[21] A. Mader and W. Eggink, "A design process for creative technology," in *Proceedings of the 16th International conference on Engineering and Product Design, E&PDE 2014* (E. Bohemia, A. Eger, W. Eggink, A. Kovacevic, B. Parkinson, and W. Wits, eds.), pp. 568–573, The Design Society, 9 2014.

[22] A. H. H., "New design methods for computer aided architectural design methodology teaching - achten henri h., 2003," Jan 2003.

[23] D. Gouwanda and S. M. N. A. Senanayake, "Emerging trends of body-mounted sensors in sports and human gait analysis," in *4th kuala lumpur international conference on biomedical engineering 2008* (N. A. Abu Osman, F. Ibrahim, W. A. B. Wan Abas, H. S. Abdul Rahman, and H.-N. Ting, eds.), pp. 715–718, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[24] K. Aminian and B. Najafi, "Capturing human motion using body-fixed sensors: outdoor measurement and clinical applications," *Computer animation and virtual worlds*, vol. 15, pp. 79–94, may 2004.

[25] N. Sun and Y. Sakai, "New approaches to human gait simulation using motion sensors," in *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 126–131, IEEE, mar 2017.

[26] G. T. Rab, "Flexible marker set for human gait analysis.," *Journal of Electromyography and Kinesiology*, vol. 1, pp. 139–145, jun 1991.

[27] G. Tao, S. Sun, S. Huang, Z. Huang, and J. Wu, "Human modeling and real-time motion reconstruction for micro-sensor motion capture," in *2011 IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems Proceedings*, pp. 1–5, IEEE, sep 2011.

[28] A. Ahmadi, F. Destelle, L. Unzueta, D. S. Monaghan, M. T. Linaza, K. Moran, and N. E. O'Connor, "3D human gait reconstruction and monitoring using body-worn inertial sensors and kinematic modeling," *IEEE sensors journal*, vol. 16, pp. 8823–8831, dec 2016.

[29] P. Caserman, P. Achenbach, and S. Gobel, "Analysis of inverse kinematics solutions for full-body reconstruction in virtual reality," *2019 IEEE 7th International Conference on Serious Games and Applications for Health (SeGAH)*, 2019.

[30] D. C. Niehorster, L. Li, and M. Lappe, "The accuracy and precision of position and orientation tracking in the htc vive virtual reality system for scientific research," may 2017.

[31] K. Isbister and F. Mueller, "Guidelines for the design of movement-based games and their relevance to hci," *Human–Computer Interaction*, vol. 30, no. 3-4, pp. 366–399, 2015.

[32] M. Slater and S. Wilbur, "A framework for immersive virtual environments (five): Speculations on the role of presence in virtual environments," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 6, p. 603–616, 1997.

[33] M. Hergaarden, “Graphics shaders,” *Graphics shaders*, jan 2011.

[34] S. Worley, “A cellular texture basis function,” in *A cellular texture basis function*, (New York, New York, USA), pp. 291–294, ACM Press, 1996.

[35] K. Suresh, “An overview of randomization techniques: An unbiased assessment of outcome in clinical research,” *Journal of Human Reproductive Sciences*, vol. 4, no. 1, p. 8, 2011.

[36] R. M. Ryan and E. L. Deci, “Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being.,” *American Psychologist*, vol. 55, no. 1, p. 68–78, 2000.

A Interviews

A.1 Richard Loos

Rowing Insights with Richard Loos, coach at Thyro - 08-03-2019

Ergometer specific (in contrast to on the water rowing)

- Low engagement
- More soreness due to lactation of muscles recovers longer
- Wrong technique feedback from the activity itself, only focus on raw pull strength and rowing distance, not on correct form
- Synchronous rowing cannot be trained
- The turning of the oar, to enter and exit the water, cannot be trained since on the ergometer the handle needs to be straight

Rowing Context

- Different boat types, a single scull, a double scull (two oars per person) and a coxed octuple scull (one oar per person), in general, the more teammates the easier
- Coxswain positioned in the back of the boat looks for direction, steers and indicates tempo.
- Training on an ergometer is possible with and without a coach
- Main goals of a coach are technique correction and motivation
- Ideal training consists of 4 / 5 km duration, at a consistent speed, except when overtaking another boat.
- Solo training sometimes every day, sometimes once a week, differs per person
- Technique correction for beginners is most needed in the first two years of being a rower.

Perfect rowing form

- The recovery phase of the stroke needs the following steps in order: *extending at arms - bending the lower back - retracting knees*, in order to keep the handle at a constant height so the 'oars' do not touch the water.
- The rower should not squeeze the handle
- Balancing of the rower should be done with the arms and not with the entire body
- The rower should hold their wrists stiff and straight
- In order to pull the handle, the arms should be hung not used actively
- A beginning rower should hold the oars/handle at a constant distance to the water
- A more advanced rower should hold the oars against the water until halfway during the stroke, then make room for turning the oar
- The term flagging means holding the oars too high

Possible technology application

- Heart Rate sensor
- Ergometer output
- Trackers on the handle
- Trackers on the seat
- Trackers on the head, shoulders or upper back

Figure 26: Minutes from a meeting with Richard Loos, former coach at rowing association Thyro Enschede

A.2 Euros

Interview with Abe Winters, student coach at D.R.V Euros - 16-10-2019

What do rowing benchmarks look like?

The standard benchmark in the sport of rowing is the 2km test, where the score is determined by the number of time rows 500 meters. This test is very exhausting so for this reason alternatives like 100 metres or one-minute tests are also widely used,

What do training schedules look like?

It differs, but common versions are 8 x 1 minutes and 4 x 8 minutes, with breaks between every sprint.

What does good rowing form look like?

The order of muscle movement is very important; The stroke consists of the use of quadriceps first, and the use of arms and legs last.
The recovery phase of the stroke is even more important as to not inhibit the oars; first the arms, then the lower back and then the legs.

What are some common mistakes?

- The handle should be at a consistent height, except when emulating rowing on the water, then the movement of inserting the oars in the water should be done.
- The movement of arms and back should be the last one
- Contract the abdominal muscles as to push your entire body to the back and the arms and upper body are not left behind
- End with a hollow back

Figure 27: Minutes from observations and interviews with Abe Winters, student coach at rowing association Euros in Enschede

B Prototype

B.1 Algorithms

B.1.1 Motion data management

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using UnityEditor.IMGUI.Controls;
using UnityEngine;
using UnityEngine.Events;

/// <summary>
/// this class combines all motion data (from different
    transform recorders) into a single instance
/// with some calculationg of back angles and distances
///
/// Virtual Rowing research project
/// Made by Koen Vogel - k.a.vogel@student.utwente.nl
/// </summary>

public class MotionDataManager : MonoBehaviour
{

    [Header("General")]
    public StatusDisplay display;

    [Header("Recording points")]
    public TransformRecorder handle;
    public TransformRecorder headset;
    public TransformRecorder seat;
    public TransformRecorder wheel;
    public TransformRecorder chest;

    //tracked distances and angles
    [HideInInspector] public List<float> backAngles = new List<
        float>();
    [HideInInspector] public List<Vector3> distWheelHandle = new
         List<Vector3>();
    [HideInInspector] public List<Vector3> distSeatWheel = new
        List<Vector3>();
    [HideInInspector] public List<Vector3> distHandleChest = new
         List<Vector3>();

    //velocities of distances (relative velocities)
    [HideInInspector] public List<Vector3> velDistWheelHandle =
        new List<Vector3>();
    [HideInInspector] public List<Vector3> velDistSeatWheel =
        new List<Vector3>();
    [HideInInspector] public List<Vector3> velDistHandleChest =
        new List<Vector3>();

    //easy access to most current distance and angle values
    [HideInInspector] public float CurrentBackAngle
```

```
    {
        get
        {
            if (backAngles.Count < 1)
                return -999;
            return backAngles[backAngles.Count - 1];
        }
    }
    [HideInInspector] public Vector3 CurrentDistWheelHandle
    {
        get
        {
            if (distWheelHandle.Count < 1)
                return Vector3.zero;
            return distWheelHandle[distWheelHandle.Count - 1];
        }
    }
    [HideInInspector] public Vector3 CurrentDistSeatWheel
    {
        get
        {
            if (distSeatWheel.Count < 1)
                return Vector3.zero;
            return distSeatWheel[distSeatWheel.Count - 1];
        }
    }
    [HideInInspector] public Vector3 CurrentDistHandleChest
    {
        get
        {
            if (distHandleChest.Count < 1)
                return Vector3.zero;
            return distHandleChest[distHandleChest.Count - 1];
        }
    }

    //easy access to most current velocity values
    [HideInInspector] public Vector3 CurrentVelDistWheelHandle { get
    {
        if (velDistWheelHandle.Count <= 3) return Vector3.zero;
        return velDistWheelHandle[velDistWheelHandle.Count - 1];
    }}
    [HideInInspector] public Vector3 CurrentVelDistSeatWheel
    {
        get
        {
            if (velDistSeatWheel.Count < 1) return Vector3.zero;
            return velDistSeatWheel[velDistSeatWheel.Count - 1];
        }
    }
    [HideInInspector] public Vector3 CurrentVelDistHandleChest { get
    {
        if (velDistHandleChest.Count <= 3) return Vector3.zero;
        return velDistHandleChest[velDistHandleChest.Count - 1];
    }}
```

```
    private void FixedUpdate()
    {
        //do not record if there is no data available
        if (handle.pos.Count <= 0) return;



        //record distances
        distWheelHandle.Add(wheel.pos[wheel.pos.Count - 1] - handle.pos[handle.pos.Count - 1]);
        distSeatWheel.Add(seat.pos[seat.pos.Count - 1] - wheel.pos[wheel.pos.Count - 1]);
        distHandleChest.Add(handle.pos[handle.pos.Count - 1] - chest.pos[chest.pos.Count - 1]);

        //calculate relative velocities between relative positions
        velDistWheelHandle.Add(CalculateVelocity(distWheelHandle));
        velDistSeatWheel.Add(CalculateVelocity(distSeatWheel));
        velDistHandleChest.Add(CalculateVelocity(distHandleChest));
    }

    private Vector3 CalculateVelocity(List<Vector3> pos)
    {
        if (pos.Count <= 3) return(Vector3.zero);
        var diff1 = pos[pos.Count - 1] - pos[pos.Count - 2];
        var diff2 = pos[pos.Count - 2] - pos[pos.Count - 3];
        var diff3 = pos[pos.Count - 3] - pos[pos.Count - 4];
        var diff = (diff1 + diff2 + diff3) / 3;
        var scale = 1 / Time.fixedDeltaTime;
        return(diff * scale);
    }

    private void Update()
    {

        //visualize speeds to display
        display.handleSpeedRelative.text = "CurVelDistWheHand: " + CurrentVelDistWheelHandle.z.ToString("F");
        if (handle.pos.Count > 0 && distHandleChest.Count > 0 && distSeatWheel.Count > 0 && backAngles.Count > 0)
        {
            display.UpdateSpeed(Mathf.Abs(handle.Velocity.z), seat.Velocity.z, Mathf.Abs(headset.Velocity.z),
                Mathf.Abs(chest.Velocity.z), distSeatWheel[distSeatWheel.Count - 1].z,
                distHandleChest[distHandleChest.Count - 1].z);
            display.angleBack.text = "Angle of Back: " + backAngles[backAngles.Count - 1];
        }
    }

    public void ClearRecordedData()
    {
```

```

        //resetting recorders
        handle.ClearData();
        headset.ClearData();
        seat.ClearData();
        wheel.ClearData();
        chest.ClearData();

        //clearing lists
        backAngles.Clear();

        distWheelHandle.Clear();
        distSeatWheel.Clear();
        distHandleChest.Clear();

        velDistWheelHandle.Clear();
        velDistSeatWheel.Clear();
        velDistHandleChest.Clear();
    }
}
```

B.1.2 Stroke detection

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using UnityEditor.IMGUI.Controls;
using UnityEngine;
using UnityEngine.Events;
using UnityEngine.UIElements;

/// <summary>
/// class responsible realtime detection of system events like stroke detection
///
/// Virtual Rowing research project
/// Made by Koen Vogel - k.a.vogel@student.utwente.nl
/// </summary>

public class StrokeController : MonoBehaviour
{
    #region variables
    //setup delegates and enums
    public delegate void EndOfStroke(StrokeData stroke);
    [HideInInspector] public EndOfStroke endOfStroke;
    public enum State { Prep, Rowing, Idle }
    public State GameState
    {
        get
        {
            if (recordedStrokes.Count <= 0) return State.Prep;
            //if (!moving) return State.Idle;
            return State.Rowing;
        }
    }
    public static StrokeController Instance { get; private set; }
    private void Awake()
    {
        if (Instance != null && Instance != this)
        {
            Destroy(this.gameObject);
        } else {
            Instance = this;
        }
    }

    [Header("References")]
    public StatusDisplay display;
    public MotionDataManager moCap;
    public Logger logger;

    [Header("Settings")]
    public Transform baseReference;
    public bool endOfStrokeOverride;
```

```
    [HideInInspector] public List<StrokeData> recordedStrokes = new List<StrokeData>();
    #endregion

    private void Update()
    {
        //visualize speeds to display
        display.gamestate.text = "StrokeState: " + strokeState.ToString();

        //trigger and record stroke if detected
        if (DetectEndOfStroke()) TriggerEndOfStroke();
    }

    //this function is called every stroke
    private void TriggerEndOfStroke()
    {
        var handle = moCap.handle.pos.GetRange(maxHandlePosIndex, moCap.handle.pos.Count - 1 - maxHandlePosIndex);
        var headset = moCap.headset.pos.GetRange(maxHandlePosIndex, moCap.handle.pos.Count - 1 - maxHandlePosIndex);
        var seat = moCap.seat.pos.GetRange(maxHandlePosIndex, moCap.handle.pos.Count - 1 - maxHandlePosIndex);
        var wheel = moCap.wheel.pos.GetRange(maxHandlePosIndex, moCap.handle.pos.Count - 1 - maxHandlePosIndex);
        var chest = moCap.chest.pos.GetRange(maxHandlePosIndex, moCap.handle.pos.Count - 1 - maxHandlePosIndex);
        var newStroke = new StrokeData(baseReference.position, handle, headset, seat, wheel, chest, DateTime.Now); //actually parse all data to container
        Debug.Log("points recorded: " + (moCap.handle.pos.Count - 1 - maxHandlePosIndex));
        endOfStroke.Invoke(newStroke);        //trigger system-wide delegate function
        Logger.Instance.SaveLog(newStroke);   //try to log stroke information to disk
        recordedStrokes.Add(newStroke);       //record to volatile memory for use while running
        moCap.ClearRecordedData();            //clear all motion variables when stroke is over
    }

    #region Real-time detection

    private enum StrokeState { Idle, Pull, Push }
    private StrokeState strokeState = StrokeState.Idle;

    //peaks
    private float minHandlePos = 999f;
    private float maxHandlePos = -999f;
    private float minSeatPos = 999f;
    private float maxSeatPos = -999f;
    private int minHandlePosIndex = 0;
    private int maxHandlePosIndex = 0;
    private int minSeatPosIndex = 0;
    private int maxSeatPosIndex = 0;
```

```
    private bool DetectEndOfStroke()
    {
        //if manually activated in inspector, trigger end of
            stroke
        if (endOfStrokeOverride)
        {
            endOfStrokeOverride = false;
            return true;
        }

        //when velocity is significant enough
        var pulling = moCap.CurrentVelDistWheelHandle.z > 0.45f;
        var pushing = moCap.CurrentVelDistWheelHandle.z < -0.4f;

        //validate stroke detection once switching velocities
            are detected
        if (pulling)
        {
            if (strokeState == StrokeState.Push)
            {
                var crit = CheckStrokeCriteria();
                if (crit)
                {
                    return true;
                }
            }
            strokeState = StrokeState.Pull;
        }
        else if (pushing)
        {
            strokeState = StrokeState.Push;
        }

        //record peaks
        if (moCap.CurrentDistWheelHandle.z > maxHandlePos)
        {
            maxHandlePos = moCap.CurrentDistWheelHandle.z;
            maxHandlePosIndex = moCap.distWheelHandle.Count - 1;
        }
        if (moCap.CurrentDistWheelHandle.z < minHandlePos)
        {
            minHandlePos = moCap.CurrentDistWheelHandle.z;
            minHandlePosIndex = moCap.distWheelHandle.Count - 1;
        }
        if (moCap.CurrentDistSeatWheel.z > maxSeatPos)
        {
            maxSeatPos = moCap.CurrentDistSeatWheel.z;
            maxSeatPosIndex = moCap.distSeatWheel.Count - 1;
        }
        if (moCap.CurrentDistSeatWheel.z < minSeatPos)
        {
            minSeatPos = moCap.CurrentDistSeatWheel.z;
            minSeatPosIndex = moCap.distSeatWheel.Count - 1;
        }

        return false;
```

```
    }

    private bool CheckStrokeCriteria()
    {
        //check if distance between peaks is enough
        var diff = Mathf.Abs(maxHandlePos - minHandlePos);
        var timeDiff = Mathf.Abs(maxHandlePosIndex -
            minHandlePosIndex) * Time.fixedDeltaTime;
        //print("diff: "+ diff);
        if (diff > 0.4f && timeDiff > 0.5f)
        {
            //clear peaks and return a correct stroke
            minHandlePos = 999f;
            maxHandlePos = -999f;
            minSeatPos = 999f;
            maxSeatPos = -999f;
            minHandlePosIndex = 0;
            maxHandlePosIndex = 0;
            minSeatPosIndex = 0;
            maxSeatPosIndex = 0;
            return true;
        }

        //check here if duration is also long enough
        return false;
    }
    #endregion

    public void ManualEndOfStrokeOverride()
    {
        Debug.Log("Manual end of stroke activated");
        TriggerEndOfStroke();
    }
}
```

B.1.3 Ghost visualizer

```
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Serialization;

/// <summary>
/// class responsible for visualizing recorded motion data in the form of a ghost character replicating the movements
///
/// Virtual Rowing research project
/// Made by Koen Vogel - k.a.vogel@student.utwente.nl
/// </summary>

public class GhostVisualizer : MonoBehaviour
{
    [Header("General")]
    public StrokeController strokeController;
    public Transform baseReference;
    private Vector3 Offset => baseReference.transform.position;

    [Header("Transforms")]
    public Transform handle;
    public Transform headset;
    public Transform seat;
    public Transform wheel;
    public Transform chest;

    public bool isVisualizing = false;

    private void Start()
    {
        strokeController.endOfStroke += VisualizeStroke;
    }

    public void VisualizeStroke(StrokeData stroke)
    {
        StartCoroutine(DisplayOverTime(stroke));
    }

    private IEnumerator DisplayOverTime(StrokeData stroke)
    {
        if (isVisualizing) yield return null;
        Debug.Log("Ghost isVisualizing stroke with data points: " + stroke.headsetPos.Count);
        isVisualizing = true;
        var offset = baseReference.position - stroke.offsetRef;
        for (int i = 0; i < stroke.headsetPos.Count; i++)
        {
            handle.position = stroke.handlePos[i] + offset;
            headset.position = stroke.headsetPos[i] + offset;
            seat.position = stroke.seatPos[i] + offset;
            wheel.position = stroke.wheelPos[i] + offset;
            chest.position = stroke.chestPos[i] + offset;
```

```
            yield return new WaitForSeconds(Time.fixedDeltaTime)
                ;
        }
        Debug.Log("Finished ghost playback");
        isVisualizing = false;
        yield return null;
    }
}
```

B.1.4 Stroke recorder & technique detection

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.InteropServices;
using Unity.Mathematics;
using UnityEngine;

/// <summary>
/// data container class; contains all important parameters and
    raw motion data for a single rowing stroke
///
/// Virtual Rowing research project
/// Made by Koen Vogel - k.a.vogel@student.utwente.nl
/// </summary>

[System.Serializable]
public class StrokeData
{
    //raw data
    public List<float> backAngles = new List<float>();
    public List<Vector3> handlePos = new List<Vector3>();
    public List<Vector3> headsetPos = new List<Vector3>();
    public List<Vector3> seatPos = new List<Vector3>();
    public List<Vector3> wheelPos = new List<Vector3>();
    public List<Vector3> chestPos = new List<Vector3>();

    //velocities
    public List<Vector3> handleVel = new List<Vector3>();
    public List<Vector3> headsetVel = new List<Vector3>();
    public List<Vector3> seatVel = new List<Vector3>();
    public List<Vector3> wheelVel = new List<Vector3>();
    public List<Vector3> chestVel = new List<Vector3>();

    //distances
    public List<float> distHandleChest = new List<float>();
    public List<float> distSeatWheel = new List<float>();

    //relative velocities
    public List<float> velDistHandleChest = new List<float>();
    public List<float> velDistSeatWheel = new List<float>();
    public List<float> velBackAngles = new List<float>();

    //maximum distances
    public float DistMaxHandleChest
    {
        get
        {
            var highest = -999f;
            foreach (var dist in distHandleChest.Where(dist =>
                dist > -highest))
            {
                highest = dist;
            }
```

```
            return highest;
        }
    }
    public float DistMaxSeatFeet
    {
        get
        {
            var highest = -999f;
            foreach (var dist in distSeatWheel.Where(dist => dist > -highest))
            {
                highest = dist;
            }
            return highest;
        }
    }

    //muscle movement timings
    public MuscleMovementTimings legMovements;
    public MuscleMovementTimings backMovements;
    public MuscleMovementTimings armMovements;

    //parameters
    public DateTime TimestampStart => timestampEnd - Duration;
    public DateTime timestampEnd;
    public float DurationSeconds => SampleSize * Time.fixedDeltaTime;
    public TimeSpan Duration => new TimeSpan(0, 0, 0, 0, Mathf.RoundToInt(DurationSeconds * 1000));
    public int SampleSize => handlePos.Count;
    public Vector3 offsetRef;

    //return the index of all motion data lists where the rowing stroke switches to the recovery phase
    public int SwitchIndex
    {
        get
        {
            var lowest = 999f;
            var lowestIndex = 0;
            for (int i = SampleSize - 1; i >= 0; i--)
            {
                if (distHandleChest[i] < lowest)
                {
                    lowest = distHandleChest[i];
                    lowestIndex = i;
                }

                if (distHandleChest[i] > lowest + 0.1f) return lowestIndex;
            }
            return lowestIndex;
        }
    }

    //contructor
```

```
    public StrokeData(Vector3 offsetRef, List<Vector3> handlePos, List<Vector3> headsetPos, List<Vector3> seatPos,
        List<Vector3> wheelPos, List<Vector3> chestPos, DateTime timeEnd)
    {
        //parameters
        this.offsetRef = new Vector3(offsetRef.x, offsetRef.y, offsetRef.z);
        timestampEnd = timeEnd;

        //positions
        foreach (var pos in handlePos){ this.handlePos.Add(pos); }
        foreach (var pos in headsetPos){ this.headsetPos.Add(pos); }
        foreach (var pos in seatPos) { this.seatPos.Add(pos); }
        foreach (var pos in wheelPos) { this.wheelPos.Add(pos); }
        foreach (var pos in chestPos) { this.chestPos.Add(pos); }

        //velocities
        for (var i = 0; i < SampleSize; i++)
        {
            handleVel.Add(GetVelocity(i, handlePos));
            headsetVel.Add(GetVelocity(i, headsetPos));
            seatVel.Add(GetVelocity(i, seatPos));
            wheelVel.Add(GetVelocity(i, wheelPos));
            chestVel.Add(GetVelocity(i, chestPos));
        }

        //distance calculator
        for (var i = 0; i < SampleSize; i++) { distHandleChest.Add( Mathf.Abs(this.handlePos[i].z - this.chestPos[i].z)); }
        for (var i = 0; i < SampleSize; i++) { distSeatWheel.Add( Mathf.Abs(this.wheelPos[i].z - this.seatPos[i].z)); }

        //relative velocities
        for (var i = 0; i < SampleSize; i++) { velDistHandleChest.Add( GetVelocity(i, distHandleChest)); }
        for (var i = 0; i < SampleSize; i++) { velDistSeatWheel.Add( GetVelocity(i, distSeatWheel)); }

        //record back angles
        for (int i = 0; i < SampleSize; i++)
        {
            var y = chestPos[i].y - seatPos[i].y;
            var z = chestPos[i].z - seatPos[i].z;
            backAngles.Add(Mathf.Abs(Mathf.Atan(y / z)));
            //backAngles.Add(chestPos[i].z - seatPos[i].z);
        }
        for (var i = 0; i < SampleSize; i++) { velBackAngles.Add( GetVelocity(i, backAngles)); }

```

```
        //generate timings for muscle order
        legMovements = new MuscleMovementTimings(GetTimings(distSeatWheel));
        armMovements = new MuscleMovementTimings(GetTimings(distHandleChest));
        backMovements = new MuscleMovementTimings(GetTimings(backAngles));
    }

    //function to extract the significant motion for the different body parts
    private float2 GetTimings(List<float> values)
    {
        //select all data from the recovery phase
        var vel = values.GetRange(SwitchIndex, SampleSize - 1 - SwitchIndex);
        if (vel[0] > vel[vel.Count - 1])
        {
            vel = vel.Select(t => -1 * t).ToList();
        }

        //determine range of significane
        var lowest = vel.ToArray().Min();
        var highest = vel.ToArray().Max();
        var diff = highest - lowest;
        var thresholdLow = lowest + (0.22 * diff);
        var thresholdHigh = lowest + 0.78 * diff;

        //determine start and end indexes of range of significance
        var startIndex = 999;
        var endIndex = 999;
        for (var index = 0; index < vel.Count; index++)
        {
            if (startIndex == 999 && vel[index] > thresholdLow)
                startIndex = index;

            if (startIndex != 999 && endIndex == 999 && vel[index] > thresholdHigh)
                endIndex = index;
        }
        if (startIndex == 999 || endIndex == 999) Debug.Log("Warning: muscle timings not correctly configured");

        //convert to seconds
        var start = startIndex * Time.fixedDeltaTime;
        var end = endIndex * Time.fixedDeltaTime;
        return new float2(start, end);
    }

    private static Vector3 GetVelocity(int i, List<Vector3> pos)
    {
        if (i <= 3) return Vector3.zero;
        var diff1 = pos[pos.Count - 1] - pos[pos.Count - 2];
        var diff2 = pos[pos.Count - 2] - pos[pos.Count - 3];
        var diff3 = pos[pos.Count - 3] - pos[pos.Count - 4];
        var diff =  (diff1 + diff2 + diff3) / 3;
```

```
        var scale = 1 / Time.fixedDeltaTime;
        return diff * scale;
    }

    private static float GetVelocity(int i, List<float> pos)
    {
        if (i <= 3) return 0;
        var diff1 = pos[pos.Count - 1] - pos[pos.Count - 2];
        var diff2 = pos[pos.Count - 2] - pos[pos.Count - 3];
        var diff3 = pos[pos.Count - 3] - pos[pos.Count - 4];
        var diff =  (diff1 + diff2 + diff3) / 3;
        var scale = 1 / Time.fixedDeltaTime;
        return diff * scale;
    }

    //data container struct for the final movement timings
    [System.Serializable]
    public struct MuscleMovementTimings
    {
        public float start;
        public float end;
        public float Duration => end - start;

        public MuscleMovementTimings(float2 timings)
        {
            this.start = timings.x;
            this.end = timings.y;
        }
    }
}
```

B.2 Feedback

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.Globalization;
using System.Numerics;
using Unity.Mathematics;
using UnityEngine;
using UnityEngine.UI;
using Vector3 = UnityEngine.Vector3;

/// <summary>
/// class responsible displaying movement timings to bar chart
///
/// Virtual Rowing research project
/// Made by Koen Vogel - k.a.vogel@student.utwente.nl
/// </summary>

public class BarController : MonoBehaviour
{

    #region variables
    #pragma warning disable 649

    [Header("References")]
    [SerializeField] private Slider slider1;
    [SerializeField] private Slider slider2;
    [SerializeField] private Slider slider3;
    [SerializeField] private TMPro.TextMeshProUGUI slider1Header;
    [SerializeField] private TMPro.TextMeshProUGUI slider2Header;
    [SerializeField] private TMPro.TextMeshProUGUI slider3Header;
    [SerializeField] private TMPro.TextMeshProUGUI windowMaxText;

    //
    private float WindowMaxX => transform.position.x + GetComponent<RectTransform>().rect.width;
    private float WindowMinX => 0;


    #pragma warning restore 649
    #endregion

    // Start is called before the first frame update
    private void Start()
    {
        //subscribe to system-wide event
        StrokeController.Instance.endOfStroke += EndOfStroke;
    }

    private void EndOfStroke(StrokeData stroke)
    {
```

```
        //extract useful information from stroke data container
        if (stroke.backAngles.Count == 0 || stroke.
            velDistSeatWheel.Count == 0 || stroke.
            velDistHandleChest.Count == 0) return;
        var legs = stroke.legMovements;
        var back = stroke.backMovements;
        var arms = stroke.armMovements;

        //start visualizing data to barchart
        DisplayBars(legs.start, legs.end, back.start, back.end,
            arms.start, arms.end);
    }

    public void DisplayBars(float bar1Start, float bar1End,
        float bar2Start, float bar2End, float bar3Start, float
        bar3End)
    {
        //find minimum and maximum
        var array = new[] { bar1Start, bar1End, bar2Start,
            bar2End, bar3Start, bar3End };
        var lowest = Mathf.Min(array);
        var highest = Mathf.Max(array);
        windowMaxText.text = highest.ToString("F");

        //remap values
        var bar1StartMapped = bar1Start.Remap(lowest, highest,
            WindowMinX, WindowMaxX);
        var bar1EndMapped = bar1End.Remap(lowest, highest,
            WindowMinX, WindowMaxX);
        var bar2StartMapped = bar2Start.Remap(lowest, highest,
            WindowMinX, WindowMaxX);
        var bar2EndMapped = bar2End.Remap(lowest, highest,
            WindowMinX, WindowMaxX);
        var bar3StartMapped = bar3Start.Remap(lowest, highest,
            WindowMinX, WindowMaxX);
        var bar3EndMapped = bar3End.Remap(lowest, highest,
            WindowMinX, WindowMaxX);

        //apply bar visuals to game objects
        ApplyBar(slider1, slider1Header.transform,
            bar1StartMapped, bar1EndMapped - bar1StartMapped);
        ApplyBar(slider2, slider2Header.transform,
            bar2StartMapped, bar2EndMapped - bar2StartMapped);
        ApplyBar(slider3, slider3Header.transform,
            bar3StartMapped, bar3EndMapped - bar3StartMapped);

    }

    private void ApplyBar(Slider slider, Transform header, float
         start, float width)
    {
        //apply position
        var pos = slider.GetComponent<RectTransform>().
            anchoredPosition;
        slider.GetComponent<RectTransform>().anchoredPosition =
            new UnityEngine.Vector2(start, pos.y);

```

```
        //move header along with bar
        var position = header.GetComponent<RectTransform>().
            anchoredPosition;
        header.GetComponent<RectTransform>().anchoredPosition =
            new UnityEngine.Vector2(start, position.y);

        //apply width of slider
        slider.value = width.Remap(WindowMinX, WindowMaxX, 0, 1)
            ;
    }
}
```

B.3 Environment

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Globalization;
using UnityEngine;

/// <summary>
/// class responsible for controlling the movement of the rowing boat (single scull)
///
/// Virtual Rowing research project
/// Made by Koen Vogel - k.a.vogel@student.utwente.nl
/// </summary>

public class ScullBehaviour : MonoBehaviour
{
    #region variables
    #pragma warning disable 649

    [Header("References")]
    [SerializeField] private StrokeController controller;
    [SerializeField] private TransformRecorder handle;
    [SerializeField] private TMPro.TextMeshProUGUI velocityText;
    [SerializeField] private Transform movingPart;

    [Header("Settings")]
    [SerializeField] private Vector3 direction;

    private float HandleSpeed => -controller.moCap.CurrentVelDistWheelHandle.z;
    private static float _currentVelocity = 0;
    [SerializeField] private float accelerationMultiplier;
    [SerializeField] private bool enableQuadraticDrag;
    [SerializeField] private float quadDragExponent;
    [SerializeField] private float quadDragMultiplier;
    [SerializeField] private float linearDrag;

    #pragma warning restore 649
    #endregion


    // Update is called once per frame
    private void FixedUpdate()
    {
        //move boat
        movingPart.Translate((Time.fixedDeltaTime * _currentVelocity) * direction);

        //apply acceleration from handle
        if (HandleSpeed < 0)
        {
            float acceleration = HandleSpeed * accelerationMultiplier;
            _currentVelocity += acceleration;
```

```
        }

        //apply drag
        var quadDrag = quadDragMultiplier * Mathf.Pow(
            _currentVelocity, quadDragExponent);
        if (enableQuadraticDrag)
            _currentVelocity += quadDrag;
        else
            _currentVelocity *= linearDrag;
    }

    private void LateUpdate()
    {
        //display velocity to text
        velocityText.text = Mathf.Abs(_currentVelocity).ToString
            ("F") + " m/s";
    }
}
```

B.4 Water shader in shader graph

Tutorials used for the implementation of the water shader:

- https://www.youtube.com/watch?v=Vg0L9aCRWPE&t=698s
- https://www.youtube.com/watch?v=jBmBb-je4Lg

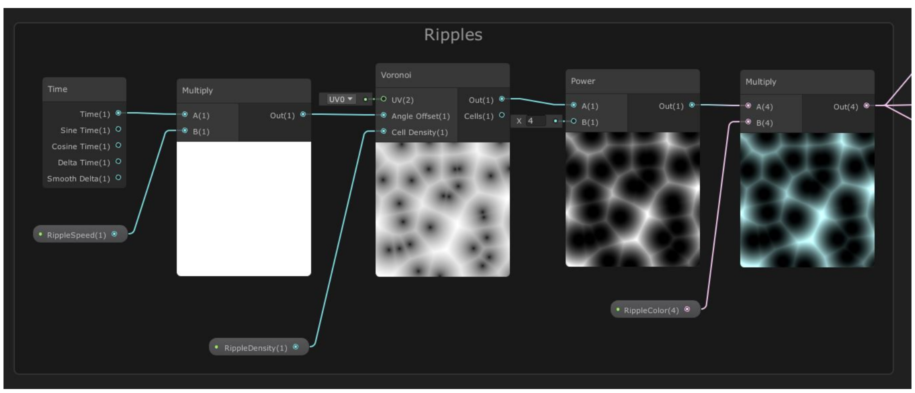

Figure 28: Ripple implementation in the water shader

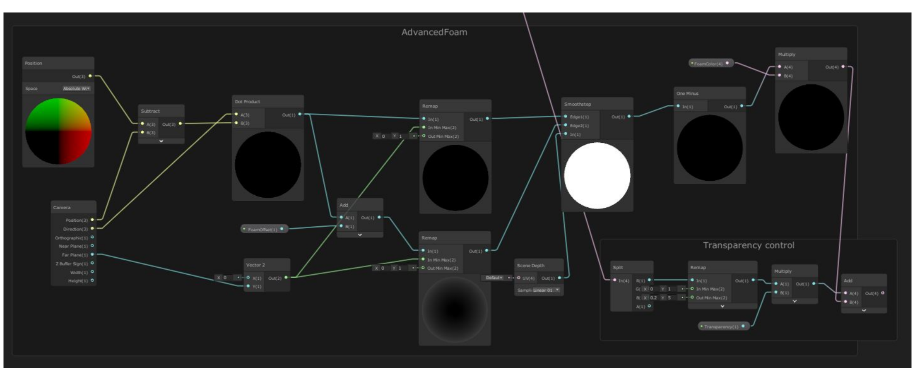

Figure 29: Foam implementation in the water shader

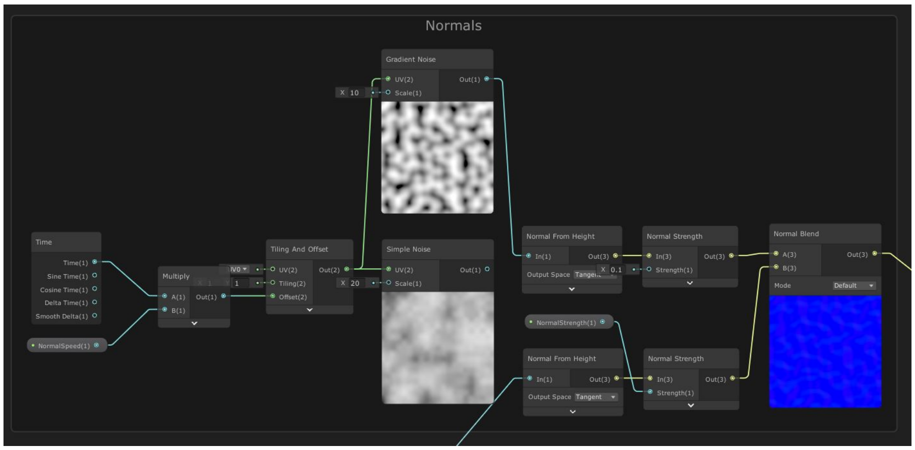

Figure 30: Reflection implementation in the water shader

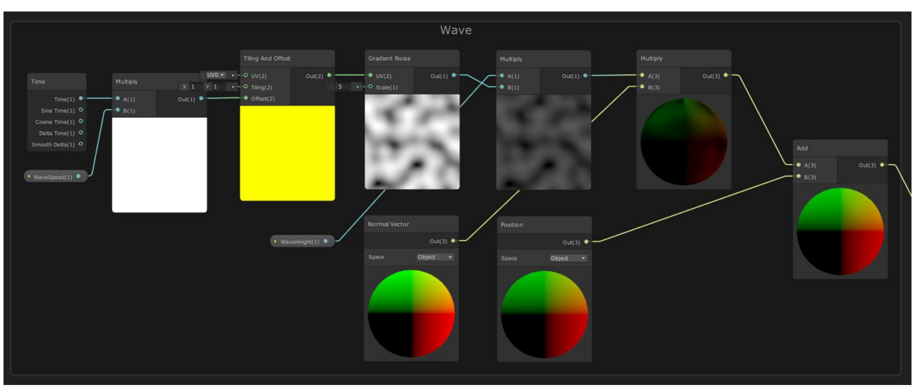

Figure 31: Wave implementation in the water shader

C User tests

C.1 Informed consent forms

PP nr.

CONSENT FORM

Project title: Virtual Rowing
Researcher: Koen Vogel

Research & Data collection explanation:
You are invited to participate in a user test about rowing technique. For this test, you will be asked to use a rowing ergometer while wearing a chest strap with a tracking device, with the goal of improving your rowing technique using a provided instruction video and mirror. Before and after the test, you will be asked to answer a few questions and/or fill in a small survey, which includes gathering limited personal identifiable data like age and gender. The goal of this research is to analyze possible technique improvement in rowing. The duration of the test is approximately 20-30 minutes. Recorded data consists only of motion tracking data of the 4 tracking devices, and provided information by the participant. This data will be stored for a maximum of 10 years, as per VSNU guidelines. The recorded material will be not be shared outside HMI and will be dealt with anonymously.

Please check the boxes.	
1. I have read the explanation and I understand that I can ask questions at any time during the test.	☐
2. I understand that I can quit at any time, without having to give a reason, and that my test will not be part of any data analyses.	☐
3. I give permission for my data to be used for the goals of this test and for future research into interactive play by HMI.	☐

____
Name participant

__________________ __________________
Signature participant Date

__________________ __________________
Date Signature researcher

Figure 32: Informed consent form type A, for user tests

PP nr.

CONSENT FORM

Project title: Virtual Rowing
Researcher: Koen Vogel

Research & Data collection explanation:
You are invited to participate in a user test about rowing technique. For this test, you will be asked to use a rowing ergometer while wearing a VR headset and chest strap with a tracking device, with the goal of improving your rowing technique using a provided instruction video and virtual feedback system. Before and after the test, you will be asked to answer a few questions and/or fill in a small survey, which includes gathering limited personal identifiable data like age and gender. The goal of this research is to analyze possible technique improvement in rowing. The duration of the test is approximately 20-30 minutes. Recorded data consists only of motion tracking data of the 4 tracking devices, and provided information by the participant. This data will be stored for a maximum of 10 years, as per VSNU guidelines. The recorded material will be not be shared outside HMI and will be dealt with anonymously. Please know that there is a small know risk of motion sickness when immersed in a Virtual Environment with moving elements. In case of physical discomfort, please stop the test by removing the headset, in which case your personal identifiable data will not be used.

Please check the boxes.	
1. I have read the explanation and I understand that I can ask questions at any time during the test.	☐
2. I understand that I can quit at any time, without having to give a reason, and that my test will not be part of any data analyses.	☐
3. I give permission for my data to be used for the goals of this test and for future research into interactive play by HMI.	☐

____
Name participant

__________________ Signature participant

__________________ Date

__________________ Date

__________________ Signature researcher

Figure 33: Informed consent form type B, for user tests

C.2 Task evaluation questionnaire

1. While I was working on the task I was thinking about how much I enjoyed it.
2. I did not feel at all nervous about doing the task.
3. I felt that it was my choice to do the task.
4. I think I am pretty good at this task.
5. I found the task very interesting.
6. I felt tense while doing the task.
7. I think I did pretty well at this activity, compared to other students.
8. Doing the task was fun.
9. I felt relaxed while doing the task.
10. I enjoyed doing the task very much.
11. I didn't really have a choice about doing the task.
12. I am satisfied with my performance at this task.
13. I was anxious while doing the task.
14. I thought the task was very boring.
15. I felt like I was doing what I wanted to do while I was working on the task.
16. I felt pretty skilled at this task.
17. I thought the task was very interesting.
18. I felt pressured while doing the task.
19. I felt like I had to do the task.
20. I would describe the task as very enjoyable.
21. I did the task because I had no choice.
22. After working at this task for awhile, I felt pretty competent.

Interest/enjoyment: 1, 5, 8, 10, 14(R), 17, 20

Perceived competence: 4, 7, 12, 16, 22

Perceived choice: 3, 11(R), 15, 19(R), 21(R)

Pressure/tension: 2(R), 6, 9(R), 13, 18

The subscale scores can then be used as dependent variables, predictors, or mediators, depending on the research questions being addressed.

Figure 34: Overview of questions from the IMI

C.3 Interview

Post-experiment Interview questions

- In what ways did you try to improve your technique?
- In what ways did the installation help with this?
- How did you recognise your rhythm?
- Where did you look the most?
- What did you think of using the installation?
- What do you think was the goal of the experiment?

C.4 Instruction

A screenshot of the instruction video provided to participants during the user tests can be seen in Figure 35. The entire video can be seen at the following link:

- *https://youtu.be/hybGN9xdjPs*

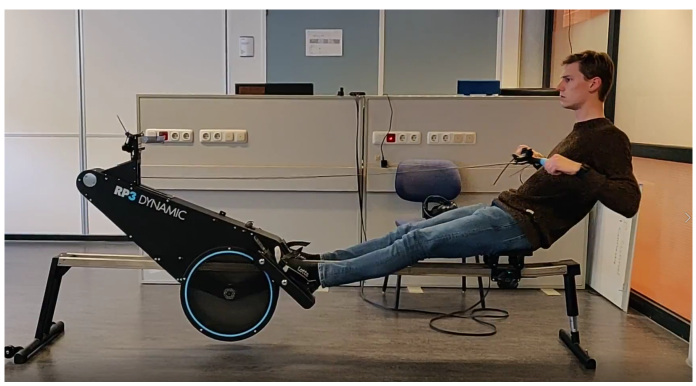

Figure 35: Screenshot from the instruction video

C.5 Complete survey

Virtual Rowing Questionnaire Full

* Required

1. Participant nr *

2. Type *

   Mark only one oval.

   - A   *Skip to question 3*
   - B   *Skip to question 8*

Pre-experiment survey type A

Thanks for participating in this experiment! This short survey will give researchers some indication about your previous experience in the field of sports and rowing.

3. What is your age?

4. What is your height? (cm)

5. What is your gender?

Mark only one oval.

- ⬭ Female
- ⬭ Male
- ⬭ Prefer not to say
- ⬭ Other: ____________________

6. How experienced are you in rowing? (boat or ergometer) *

Mark only one oval.

- ⬭ Hardly any experience
- ⬭ Occasional experience on ergometer
- ⬭ Recent professional guidance, less than 5 lessons
- ⬭ Regular professional guidance
- ⬭ Other: ____________________

7. How often do you exercise? (e.g. sports, dance etc) *

Mark only one oval.

- ⬭ Less than once a week
- ⬭ About once a week
- ⬭ Two to three times a week
- ⬭ More than three times a week

Skip to section 4 (Thanks!)

Pre-experiment survey type B

Thanks for participating in this experiment! This short survey will give researchers some indication about your previous experience in the field of sports and rowing.

8. What is your age?

____

9. What is your height?

____

10. What is your gender?

Mark only one oval.

- ◯ Female
- ◯ Male
- ◯ Prefer not to say
- ◯ Other: _______________

11. How experienced are you in rowing? (boat or ergometer) *

Mark only one oval.

- ◯ Hardly any experience
- ◯ Occasional experience on ergometer
- ◯ Recent professional guidance, less than 5 lessons
- ◯ Regular professional guidance
- ◯ Other: _______________

12. How often do you exercise? (e.g. sports, dance etc) *

Mark only one oval.

- ◯ Less than once a week
- ◯ About once a week
- ◯ Two to three times a week
- ◯ More than three times a week

13. How much experience do you have with Virtual Reality? *

Mark only one oval.

- Never tried it
- Used it 1 - 5 times
- Used it more than 5 times

Thanks!

Please return to the researcher. You will be asked to fill in a final survey after the experiment

Post-experiment survey

You will be asked a few questions about your experience using the installation. The 'task' mentioned in the questions below refers to the improving of your technique.

14. While I was working on the task I was thinking about how much I enjoyed it *

Mark only one oval.

	1	2	3	4	5	
not at all true						very true

15. I did not feel at all nervous about doing the task *

Mark only one oval.

	1	2	3	4	5	
not at all true						very true

16. I felt that it was my choice to do the task *

Mark only one oval.

	1	2	3	4	5	
not at all true	⬭	⬭	⬭	⬭	⬭	very true

17. I think I am pretty good at this task *

Mark only one oval.

	1	2	3	4	5	
not at all true	⬭	⬭	⬭	⬭	⬭	very true

18. I found the task very interesting *

Mark only one oval.

	1	2	3	4	5	
not at all true	⬭	⬭	⬭	⬭	⬭	very true

19. I felt tense while doing the task *

Mark only one oval.

	1	2	3	4	5	
not at all true	⬭	⬭	⬭	⬭	⬭	very true

20. I think I did pretty well at this activity, compared to others *

Mark only one oval.

	1	2	3	4	5	
not at all true						very true

21. Doing the task was fun *

Mark only one oval.

	1	2	3	4	5	
not at all true						very true

22. I felt relaxed while doing the task *

Mark only one oval.

	1	2	3	4	5	
not at all true						very true

23. I enjoyed doing the task very much *

Mark only one oval.

	1	2	3	4	5	
not at all true						very true

24. I did not really have a choice about doing the task *

Mark only one oval.

	1	2	3	4	5	
not at all true	◯	◯	◯	◯	◯	very true

25. I am satisfied with my performance at this task *

Mark only one oval.

	1	2	3	4	5	
not at all true	◯	◯	◯	◯	◯	very true

26. I was anxious while doing the task *

Mark only one oval.

	1	2	3	4	5	
not at all true	◯	◯	◯	◯	◯	very true

27. I thought the task was very boring *

Mark only one oval.

	1	2	3	4	5	
not at all true	◯	◯	◯	◯	◯	very true

28. I felt like I was doing what I wanted to do while I was working on the task *

Mark only one oval.

	1	2	3	4	5	
not at all true	◯	◯	◯	◯	◯	very true

29. I felt pretty skilled at this task *

Mark only one oval.

	1	2	3	4	5	
not at all true	◯	◯	◯	◯	◯	very true

30. I thought the task was very interesting *

Mark only one oval.

	1	2	3	4	5	
not at all true	◯	◯	◯	◯	◯	very true

31. I felt pressured while doing the task *

Mark only one oval.

	1	2	3	4	5	
not at all true	◯	◯	◯	◯	◯	very true

32. I felt like I had to do the task *

Mark only one oval.

	1	2	3	4	5	
not at all true	⬭	⬭	⬭	⬭	⬭	very true

33. I would describe the task as very enjoyable *

Mark only one oval.

	1	2	3	4	5	
not at all true	⬭	⬭	⬭	⬭	⬭	very true

34. I did the task because I had no choice *

Mark only one oval.

	1	2	3	4	5	
not at all true	⬭	⬭	⬭	⬭	⬭	very true

35. After working at this task for awhile, I felt pretty competent *

Mark only one oval.

	1	2	3	4	5	
not at all true	⬭	⬭	⬭	⬭	⬭	very true

D Analysis

D.1 Data processing

Listing 1: Algorithm for importing and parsing logs, and extracting useful information

```

clear
% CHECK handmatig of er niet twee pauzes zijn....

master_path = 'C:/Users/Vogel/Documents/GP_unity/Assets/Logs';
% Get a list of all files and folders in this folder.
logs = dir(master_path);
% Get a logical vector that tells which is a directory.
dirFlags = [logs.isdir];
% Extract only those that are directories.
subFolders = logs(dirFlags);
% Print folder names to command window.
combined_results = zeros(length(subFolders), 1);
count_results = 1;

for k = 3 : length(subFolders)

    %clear variables
    clear af_outliers b4_outliers dataafter databefore fil_af fil_bf files gaps_af gaps_bf string seq_b4 seq_af

    %find list of logs & setup matrix
    subfolder_path = append(master_path, '/', subFolders(k).name);
    logs = dir(fullfile(subfolder_path, '*.txt'));
    nr_of_files = length(logs);
    databefore = zeros(3,2,4); % Arm/Back/Leg - Start/End - Stroke nr

    is_after = 0;
    for file = 1:nr_of_files
        filename = logs(file).name;
        fileL = length(filename);
        count_streepje = 0;

        %find the minute property of file name
        for i = 1:fileL
            if filename(i) == '-'
                count_streepje = count_streepje + 1;
```

```
                if count_streepje == 3
                    time_start = i + 1;
                elseif count_streepje == 4
                    time_end = i - 1;
                end
            end
        end

        %determine the differences in minutes of consecutive
            logs
        if time_start == time_end
            time_new = 0;
        else
            time_new = str2double(filename(time_start:time_end))
                ;
        end
        if file == 1
            time_old = time_new;
        end
        if time_new - time_old < 0
            time_new = time_new + 60;
        end
        if is_after && time_new - time_old >=3
            disp('Error: two breaks found')
        end

        %determine where the break is between logs
        if ~is_after && time_new - time_old >= 3 && file > 1
            is_after = 1;
            afternr = file;
            dataafter = zeros(3,2,nr_of_files-file);
            disp(append('Timegap ', subFolders(k).name , ': ',
                num2str(file)));
        end
        time_old = time_new;

        %parse string to numbers and put in matrix
        string = fileread(append(subfolder_path,'/',filename));
        N = length(string);
        count = 0;
        for i = 1:N
            if string(i) == ':'
                count = count + 1;
                if count > 4
                    finding = true;
                    countNr = 2+i;
```

```
                    while finding
                        if string(countNr) ==  newline
                            finding = false;
                        else
                            countNr = countNr + 1;
                        end
                    end
                    nr = str2double(string(i+2:countNr-1));
                    if ~is_after
                        databefore(floor((count-3)/2), 2-rem(count,2),file) = nr;
                    else
                        dataafter(floor((count-3)/2), 2-rem(count,2),file-afternr+1) = nr;
                    end
                end
            end
        end
    end

    %determine outliers
    Nbefore = afternr-1;
    Nafter = nr_of_files - afternr+1;
    b4dura = zeros(1,Nbefore); afdura = zeros(1,Nafter);
    for i = 1:Nbefore
        b4dura(i) = max(max(databefore(:,:,i))) - min(min(databefore(:,:,i)));
    end
    b4_outliers = isoutlier(b4dura, 'quartiles');
    for i = 1:Nafter
        afdura(i) = max(max(dataafter(:,:,i))) - min(min(dataafter(:,:,i)));
    end
    af_outliers = isoutlier(afdura, 'quartiles');

    %filter out outliers
    %dur_fil_b4 = b4dura(~b4_outliers);
    %dur_fil_af = afdura(~af_outliers);
    %fil_b4 = databefore(:,:,~b4_outliers);
    %fil_af = dataafter(:,:,~af_outliers);
    dur_fil_b4 = b4dura;
    dur_fil_af = afdura;
    fil_b4 = databefore;
    fil_af = dataafter;

    Nbefore = length(dur_fil_b4);
```

```
    Nafter = length(dur_fil_af);

    %gaps between bars before the practice period
    gaps_b4 = zeros(Nbefore,1);
    seq_b4 = zeros(Nbefore,1);
    for i = 1:Nbefore
        gap1_b4 = ((fil_b4(2,1,i)-fil_b4(1,2,i))/dur_fil_b4(i));
        gap2_b4 = ((fil_b4(3,1,i)-fil_b4(2,2,i))/dur_fil_b4(i));
        gap3_b4 = ((fil_b4(3,1,i)-fil_b4(1,2,i))/dur_fil_b4(i));
        gaps_b4(i) = (gap1_b4 + gap2_b4+ gap3_b4)/3;
        if (fil_b4(2,1,i) > fil_b4(1,1,i)) && (fil_b4(3,1,i) > fil_b4(2,1,i))
            seq_b4(i) = 1;
        else
            seq_b4(i) = 0;
        end
    end
    mean_gaps_b4 = mean(gaps_b4);
    mean_seq_b4 = mean(seq_b4);

    %gaps between bars after the practice period
    gaps_af = zeros(Nafter,1); % Stroke nr - (gap arms-back / gap back-legs / gap arms-legs)
    seq_af = zeros(Nafter,1);
    for i = 1:Nafter
        gap1_af = ((fil_af(2,1,i)-fil_af(1,2,i))/dur_fil_af(i));
        gap2_af = ((fil_af(3,1,i)-fil_af(2,2,i))/dur_fil_af(i));
        gap3_af = ((fil_af(3,1,i)-fil_af(1,2,i))/dur_fil_af(i));
        gaps_af(i) = (gap1_af + gap2_af + gap3_af)/3;
        if (fil_af(2,1,i) > fil_af(1,1,i)) && (fil_af(3,1,i) > fil_af(2,1,i))
            seq_af(i) = 1;
        else
            seq_af(i) = 0;
        end
    end
    mean_gaps_af = mean(gaps_af);
    mean_seq_af = mean(seq_af);

    %calculate difference in technique level
    gap_improvements = (mean_gaps_af - mean_gaps_b4)*100;
    seq_improvement = (mean_seq_af - mean_seq_b4)*100;
    improvement_score = (gap_improvements); % + 3 * seq_improvement) / 4;

```

```
    %disp(append('Improvement = ', subFolders(k).name, ', gaps:', num2str(mean(gap_improvements)), ', seq:', num2str(mean(seq_improvement))))
    disp(append('Improvement = ', subFolders(k).name, ': ' , num2str(improvement_score)))
    %disp(append(subFolders(k).name, ', seq_b4: ' , num2str(mean_seq_b4), ', seq_af: ', num2str(mean_seq_af), ', Nbefore: ', num2str(Nbefore), ', Nafter: ', num2str(Nafter)))
    combined_results(count_results) = mean(improvement_score);
    count_results = count_results + 1;
end
disp(append('Total average: ', num2str(mean(combined_results))));
```

D.2 Questionnaire results

Type	Wh	I d	I fe	I th	I fo	I fe	I th	Do	I fe	I er	I di	I ar	I w	I th	I fel	I fel	I th	I fe	I fe	I w	I di	Aft	Interest/enjoyment	Perceived competence	Perceived choice	Pressure/tension
A	3	4	5	3	5	1	2	5	1	5	5	3	4	5	5	3	5	1	5	5	5	4	94	60	100	44
A	4	5	5	4	4	2	3	4	3	4	5	3	1	5	5	2	4	1	5	4	5	4	83	64	100	48
A	2	4	5	4	3	3	5	2	3	2	5	5	2	4	4	4	3	2	5	3	5	4	54	88	96	56
A	3	3	5	4	4	3	2	4	2	3	4	4	2	4	4	3	4	2	4	4	4	4	74	68	84	48
A	1	4	5	5	4	3	4	4	4	3	5	5	1	4	4	4	2	2	5	3	5	5	60	92	96	56
A	4	1	5	4	5	2	4	5	1	5	5	4	2	5	4	3	4	3	5	5	5	3	94	72	96	36
A	2	2	4	2	3	4	2	2	3	2	4	3	3	4	2	2	3	2	2	2	4	2	51	44	64	56
A	4	3	4	4	2	3	2	4	3	2	5	3	1	4	3	2	3	2	5	3	5	3	63	56	88	48
A	3	1	5	3	5	2	4	5	3	4	5	4	1	5	4	4	4	1	5	4	5	4	86	76	96	32
A	4	1	5	4	5	1	3	5	4	4	5	5	1	5	4	4	4	3	4	5	5	5	91	84	92	40
																							75	70	91	46
B	2	2	4	2	3	3	2	4	3	4	5	2	2	4	3	2	4	3	5	3	5	2	69	40	88	52
B	4	5	5	3	4	2	3	4	2	4	5	5	1	5	5	4	5	1	5	4	5	5	86	80	100	44
B	1	2	4	2	3	1	3	4	2	2	5	4	1	4	2	1	3	1	5	3	5	2	57	48	84	28
B	4	1	5	3	5	2	3	5	2	5	5	4	1	5	5	4	5	1	5	5	5	5	97	76	100	28
B	2	1	4	4	5	2	3	5	1	5	5	4	1	4	5	4	5	1	5	5	5	4	89	76	96	24
B	4	1	5	4	3	3	3	4	4	4	5	3	1	4	4	4	4	1	5	4	5	3	77	68	96	40
B	4	2	4	3	4	2	3	4	2	5	4	4	2	5	3	3	4	2	4	4	4	4	86	68	76	40
B	1	1	5	3	4	1	3	4	2	4	5	5	1	5	5	4	4	1	4	4	5	3	74	72	96	24
B	4	2	5	3	4	2	3	4	2	4	5	4	2	4	4	3	4	2	4	4	5	4	80	68	92	40
B	3	1	5	4	4	2	3	4	2	4	5	2	3	4	4	3	4	3	4	3	3	4	74	64	84	44
																							79	66	91	36

Figure 36: Answer per participant for all 22 questions. Grey means averaged scores per participant type and subscale. Red means inversely scored

E Statistical analysis

E.1 Questionnaire

E.1.1 Reliability

Reliability Statistics

Cronbach's Alpha	N of Items
.896	7

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item-Total Correlation	Cronbach's Alpha if Item Deleted
Q1	24.0000	18.632	.436	.923
Q5	23.0000	17.895	.743	.876
Q8	22.8500	18.134	.744	.876
Q10	23.2000	15.853	.852	.861
Q14	22.5000	20.895	.665	.892
Q17	23.0500	18.471	.763	.875
Q20	23.1000	17.042	.893	.858

Figure 37: Cronbach's alpha score for the *Interest/enjoyment* subscale, with scores for if subquestions would be deleted

Reliability Statistics

Cronbach's Alpha	N of Items
.824	5

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item-Total Correlation	Cronbach's Alpha if Item Deleted
Q4	13.6500	8.661	.519	.815
Q7	14.0500	8.576	.565	.804
Q12	13.2500	7.776	.595	.796
Q16	13.9000	7.253	.739	.751
Q22	13.3500	7.292	.679	.770

Figure 38: Cronbach's alpha score for the *Perceived competence* subscale, with scores for if subquestions would be deleted

Reliability Statistics

Cronbach's Alpha	N of Items
.751	5

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item-Total Correlation	Cronbach's Alpha if Item Deleted
Q3	18.1000	4.200	.470	.728
Q11	17.9500	4.155	.694	.694
Q15	18.8500	2.661	.576	.715
Q19	18.2500	3.145	.596	.676
Q21	18.0500	3.945	.494	.717

Figure 39: Cronbach's alpha score for the *Perceived choice subscale*, with scores for if subquestions would be deleted

Reliability Statistics

Cronbach's Alpha	N of Items
.300	5

Item-Total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item-Total Correlation	Cronbach's Alpha if Item Deleted
Q2	8.0500	4.155	.067	.380
Q6	8.1500	4.345	.436	.018
Q9	7.9000	5.147	.120	.274
Q13	8.7000	5.589	.048	.328
Q18	8.6000	5.411	.144	.258

Figure 40: Cronbach's alpha score for the *Pressure/tensions* subscale, with scores for if subquestions would be deleted

E.2 Technique improvement

Descriptives

	Type			Statistic	Std. Error
improvement_score	A	Mean		24,9490	7,00742
		95% Confidence Interval for Mean	Lower Bound	9,0971	
			Upper Bound	40,8009	
		5% Trimmed Mean		24,6333	
		Median		22,4950	
		Variance		491,040	
		Std. Deviation		22,15942	
		Minimum		-4,02	
		Maximum		59,60	
		Range		63,62	
		Interquartile Range		39,79	
		Skewness		,359	,687
		Kurtosis		-1,246	1,334
	B	Mean		21,4400	11,59714
		95% Confidence Interval for Mean	Lower Bound	-4,7945	
			Upper Bound	47,6745	
		5% Trimmed Mean		22,5539	
		Median		28,4550	
		Variance		1344,936	
		Std. Deviation		36,67337	
		Minimum		-48,78	
		Maximum		71,61	
		Range		120,39	
		Interquartile Range		55,37	
		Skewness		-,638	,687
		Kurtosis		-,138	1,334

Figure 41: Descriptive statistical properties of the two groups

E.2.1 Normality test

Tests of Normality

	Type	Kolmogorov-Smirnov[a] Statistic	df	Sig.	Shapiro-Wilk Statistic	df	Sig.
improvement_score	A	,188	10	,200*	,935	10	,500
	B	,173	10	,200*	,957	10	,749

*. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

Figure 42: Normality test results

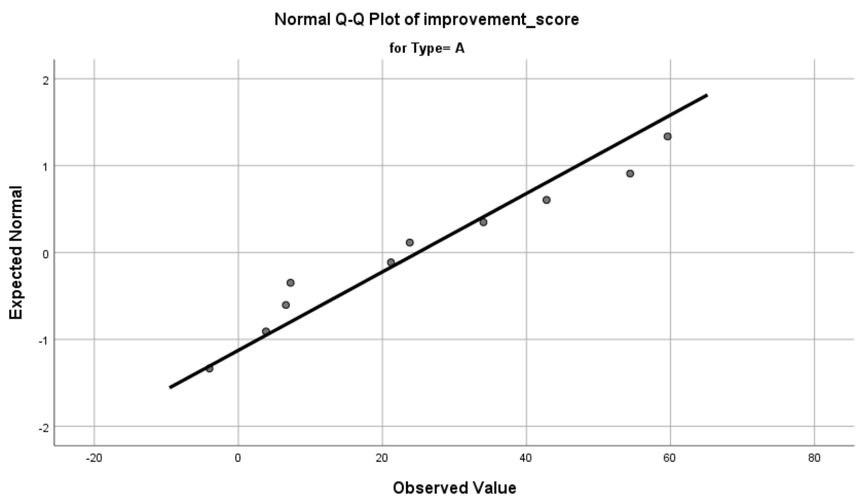

Figure 43: Q-Q plot with all data points from group A

Figure 44: Q-Q plot with all data points from group B

E.2.2 T-test

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
improvement_score	Equal variances assumed	2,499	,131	,259	18	,799	3,50900	13,54982	-24,95811	31,97611
	Equal variances not assumed			,259	14,799	,799	3,50900	13,54982	-25,40598	32,42398

Figure 45: Results from an independent samples T-test

Group Statistics

	Type	N	Mean	Std. Deviation	Std. Error Mean
improvement_score	A	10	24,9490	22,15942	7,00742
	B	10	21,4400	36,67337	11,59714

Figure 46: Comparison of statistical properties of the two groups