Western Norway
University of
Applied Sciences

BACHELOR'S THESIS

Balance Testing with Nintendo Wii Balance Board

Balansetesting med Nintendo Wii balansebrett

David Hyldgaard

Frederik Jarosch von Schweder

Bachelor, Computer Engineering

DAT190

Department for Computer Science, Electrical Engineering and Mathematical Sciences

Supervisor: Ilona Heldal

Submission date: 02/06/2020

Western Norway University of Applied Sciences

Faculty of engineering and science

Department of Computer Science, Electrical Engineering and Mathematical Sciences

Report title: Balance Testing with Nintendo Wii Balance Board	*Date:* 02.06.2020
Author(s): David Hyldgaard, Frederik Jarosch von Schweder	*Number of pages:* 42 *Number of appendix pages:* 6
Education programme: Computer Engineering	*Number of CDs/DVDs:* 0
Contact person at HVL: Ilona Heldal	*Classification:* None
Remarks: This project took place during the COVID-19 pandemic.	

External company/contact: Nasjonal Kompetansetjeneste for Vestibulære Sykdommer, Haukeland Universitetssykehus, NVKS	*External contact reference:*
External contact person: Torbjørn Aasen	*Phone:* +47 917 50 384

Summary:

This project is centered around a new algorithm called “the turning point for maintaining balance. This solution consists of a desktop application that uses a Nintendo Wii balance board to gather accurate measures for further calculations. The solution gives accurate measures on a person’s balance in the form of curve length, area, Center of Pressure (COP) and “turning point”.

This report will go in depth into the development of the application and the corresponding research and technologies to make it possible.

Keywords:

Nintendo Wii Balance Board	“The turning point for maintaining balance”	Desktop application
Open-source	Java	

Western Norway University Of Applied Sciences, Faculty of Engineering and Science

Postal address: Postboks 7030, 5020 BERGEN Address: Inndalsveien 28, Bergen

Phone: 55 58 75 00 Fax: 55 58 77 90 E-mail: post@hvl.no Home page: http://www.hvl.no

PREFACE

We would like to thank Torbjørn Aasen and Mari Kalland Knapstad from Haukeland University Hospital’s National Unit for Vestibular Diseases for trusting us with the development of this application and for helping us through the process. It has been a great learning experience and a lot of fun to work with you. Without your expertise and help it would not have been possible to complete the project this well.

We would also like to thank Ilona Heldal for excellent guidance throughout this project, especially with this report. It has been a joy to work with you and your inputs have been great. We are proud of the end product and hope it can be of help to anyone interested in this project and its potential future usages.

This project did have an extra dimension of difficulty because of the COVID-19 pandemic lockdown in Norway during the main part of this project, so especially thank you to all involved for going above and beyond in a challenging time.

TABLE OF CONTENT

1 INTRODUCTION

There are several problems and diseases influencing our sense of balance. To determine "postural sway", the horizontal movement around the center of gravity, is important for functions such as walking or breathing and helping specialists to treat patients with eventual balance problems. Careful measurements of balance play an important role in the diagnosis of several physical diseases (Lou et al., 2018).

A way to measure balance is by running tests where the patient stands on a pressure plate that measures the COP (centre of pressure), which on Wikipedia is defined as "*the point of application of the ground reaction force vector. The ground reaction force vector represents the sum of all forces acting between a physical object and its supporting surface*"[1], the movement of the COP can give insight to how a patient sways.

Tests are typically 10-60 seconds long and the result is a COP-graph, which can be compared to a random walk graph. From this data it is possible to derive parameters such as the curve length, distance traveled on the x- and y-axis, and the area of a 95% confidence ellipse. All these parameters are informative about a patient's balance, but the problem is that they all require interpretation that typically needs to be done by specialists. A newfound algorithm uses the same data to calculate *the turning point for maintaining balance*, an objective parameter that gives insight into a patient's ability to respond to critical sway. The critical sway is point indicating the postural stability when the patient will fall if this sway is ignored.

This project is based on the article "The Turning Point for Maintaining Balance" (Aasen, 2019) where an algorithm is presented with the purpose of presenting an objective evaluation of a person's balance, with concrete measurements and without the need for subjective assessments of the experts. Torbjørn Aasen then contacted Western Norway University of Applied Sciences with the wishes to implement the algorithms showed in his article and demonstrate its correctness by using easily accessible technologies.

Today, at Aasen's laboratories one will find a lot of big and expensive equipment used to measure balance and diagnose patients. These machines,however, have some obvious limitations by being stationary, large and expensive, and having complex features that keep them away from mainstream usage and from usage outside the premises of Haukeland. Aasen and his colleagues consider this a limitation in further research on defining and using measurements and tests around the turning point.

In January 2020 Torbjørn Aasen presented his idea of using a Wii Balance Board as a cheap platform to conduct balance tests and requested help for building a program which could do this. The suggestion included the calculation of the turning point based on a new algorithm - hence this project was born.

[1] Center of pressure (terrestrial locomotion), accessible from: https://en.wikipedia.org/wiki/Center_of_pressure_(terrestrial_locomotion) (accessed 14.05.2020)

With the success of this BSc project several other experts interested in insight into balance can use the results. The project could for example help special pedagogists gain insight into kid's balance and potentially catch postural complications early. Problems in balance are often correlated with vision or hearing problems too (Haanson et al, 2010). It could also help scientists to study balance outside of laboratories, since the solution is cheap, portable and useful.

1.1 Motivation and goal

Research tied to balance is quite limited to machines with their own enclosed software or programs with limited usability and often tailored to help specific investigations. These machines are often expensive and stationary, placed in laboratories, and prohibit broader usage. While there are open-source software available,many of their key features are missing to be usable, such as possibility to generate user defined tests or allow cross platform deployment.

This report presents the development of a program that can gather data from a force measuring platform, compute calculations on the data and display the results in an understandable way for balance experts but also non-experts. It is desired that the solution is cheap, accessible, and usable and understandable even for non-experts.The main goal is to develop an application that can be used for further research on balance, but also to make balance testing available for broader usage. The users should be able to gather data from a force measuring platform and then display the test results giving insight to a person's postural stability.

The initial requirements for this work is to display data with computations on curve length, area, the "turning-point" and COP-measurements (Center of Pressure). The application should make it easy for the user to define the type of test that is to be run. When a test is completed there should be a way to save or export the results as a "*.csv" file which makes it possible to follow the progress of a person over repeated tests, but also to use this data and compare with results from different people or groups.

A subgoal for the team itself is to develop a prototype early in the project. By this way the team can develop several iterations upon feedback and assure the quality of the product.

1.2 Context

The project is being conducted in cooperation with Torbjørn Aasen from Haukeland University Hospital in the department National Competence service for Vestibular Diseases. In 2019 he published the article "The Turning Point for Maintaining Balance" which in turn has led to the desire of a software that can further be used to research this newly identified "Turning point" in balance. Tied to his department, there already exists many complex machines to measure balance with enclosed software. The limitations of these machines prompted a desire for a cheap and accessible solution with open-source software, which can support further research on balance.

Following the need for an accessible and cheap solution has led to the idea of using a Wii balance board as a measuring instrument and develop a compatible program where the user can run self-defined tests. The program should then display parameters such as curve length, area of something and the turning point for maintaining balance. Which is a crucial element of the project as it hasn't been done before. The Wii balance board is a cheap platform (approx 250 NOK) compared to the machines in Haukeland's laboratories (approx 250.000 NOK), plus it's easy to transport with its size and weight (approx 1.2kg). This opens a new door that makes balance tests easy while using a lightweight, mobile and cheap alternative to the existing way to measure balance today (see Figure 1).

Figure 1.2: Wii Balance Board (left): 250 NOK, and a Haukeland machine (right): +250.000 NOK

1.3 Limitations

The main limitation was the testing of the application. Due to the COVID-19 pandemic the team was not able to gain access to Haukeland's premises and therefore not able to use the laboratory equipment. Testing of the application was therefore limited to tests that could be done remotely, which came down to software unit tests and some remote user tests where the product owner tried the application under supervision of the team.

A limitation the team did encounter was to set up the bluetooth functionality across different operating systems. The Wii Balance Board is from 2007 and hasn't been updated since. This means that the application needs to implement JSR-82, which is a bluetooth protocol for Java, and the best library to do this is BlueCove. This library is open sourced and good, but works only on 32-bit operating systems or with special drivers in place. The team did manage to get it to work on Linux consistently, even on a 64-bit version of Linux Ubuntu, but not on Windows 10 or Mac OS 10.15. This limits the usability of the application, but since it works on linux, a free and open-source operating system, anyone is able to get the application to work on most devices.

Bluetooth is essential to the application so a potential untested solution could be a bluetooth dongle with a compatible driver. With this in mind it is also important to notice that the Wii Balance Board itself may be considered a limitation to this project because of its age and limiting software compatibility to newer operating systems, but since it is well tested and is considered to be a reliable, cheap and available piece of hardware it is still the best solution so far.

1.4 Balance and measurements

Measuring human balance is a complicated field in medicine and there are many parameters and potential visualisations to consider. The desired program should focus on user defined test cases where one test consists of a test type and a time interval. A test is continuous measuring of COP data in a time interval, typically during 10-60 seconds. A test will result in a COP-graph (Image B from Figure 2) which can be compared to a random walk graph. From these measurements it is possible to calculate curve length, x- and y-curve length, area and also the turning point for maintaining balance.

All these parameters are digits and they all indicates the sway and balance of a person. The area is a 95% confidence ellipse illustrates the span of the patient's sway. If a person moves with a big movements the area will be larger compared to a person standing more steady. x and y curve length indicates the direction of the movement. If x curve length is large and y is low the patient have moved more from left to right than back and forward. x-axis movement is also called medial/lateral (ML for short) movement and y-axis movement is anterior/posterior (AP for short). Finally the total curve length gives insight into how much the patient swayed during the test. A large curve length means that the patient moved a lot.

The Turning Point for Maintaining Balance is a more sophisticated parameter which says shows the patient's reaction time to a critical sway. If a patient does not react to this critical sway, he or she will fall. This is why it can be considered a more objective parameter because it can give meaning on its own. Area and curve length must be interpreted in the context of each other and the test type, but a person's ability to react to critical sway can be meaningful without much further context.

The project is centered around Nintendo Wii balance board (Image A from Figure 2) as hardware to measure data and generate COP readings. If in the future it is desired to use other hardware to measure data, there will be no trouble to implement this in the source code. The Wii Balance Board works essentially the same way as a bathroom scale with 4 pressure sensors which measures at a rate of approximately 100Hz.

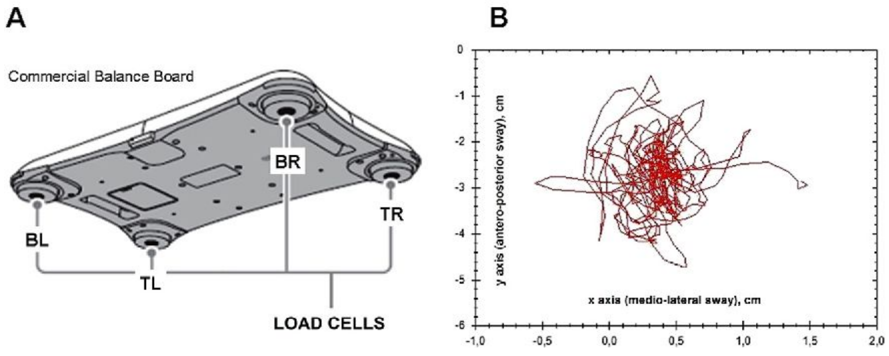

Figure 1.4, Image A shows how the Wii board is set up and image B is a COP plot from a typical test (source:https://www.semanticscholar.org/paper/We-Measure%3A-Toward-a-low-cost-portable-for-patients-Castelli-Stocchi/7a7bd417774693f71d6e73ff57aaa2b072f818af/figure/0)

1.5 Resources

To solve the problems, our team will need knowledge of software development with the primary focus being on Java development. There is also a need for skills regarding balance, especially on how to measure and compute the desired values with respect to balance. The team should also be able to provide good documentation and a good report that concludes the project. To avoid deadlines being overdue the team will need a good project lead.

Technical resources needed (and further discussed in chapter 4.3):

- The program will be developed in Java and the library for graphical user interface JavaFX.
- The project is a Maven project to simplify structure.
- Version control through Git.
- The Integrated development environment (IDE) being used is Intellij from JetBrains.

The hardware needed to complete the project is a Nintendo Wii Balance Board and 2 computers for development and writing. The team has been supplied with one Wii Balance Board from the project owner and are using their own computers to develop.

1.6 Organization of the report

Chapter 1 presents goals, problem, limitations and resources necessary are all presented in the chapter. Project owner and general context are both also introduced.

Chapter 2 presents the project background. The report discusses why this project exists, previous work leading up to this project and requirements for the finished solution. What this project means for the project owner will also be mentioned.

Chapter 3 is a short chapter on methodology that explains how the group will achieve the goal.

Chapter 4 introduces project design. This chapter discusses potential solutions, the chosen solution, tools and resources that help the group with development. Project development method, project plan, risk management and evaluation method will also be part of this chapter.

Chapter 5 discusses detailed design behind the application. The important underlying architecture, logic and User Interface will be thoroughly explained.

Chapter 6 explains the evaluation methods used, while also presenting the evaluation results.

Chapter 7 is a short chapter discussing the results following the application. These results include observations made and potential solutions to obstacles.

Chapter 8 is a discussion of the project's results and how choices were made. This chapter also includes suggestions to future use and further development.

Chapter 9 is the conclusion with a short reminder of the goals, the results and report in general

2 PROJECT BACKGROUND

This project aims to further research on balance and revolves around previous work from Torbjørn Aasen's article "The turning point for maintaining balance", which in conclusion is an algorithm that is used for diagnostic and follow-up on patients. The project also aims to make balance testing available to any interested party, which means that calculations only available by using expensive machines today can be available by a low-cost solution.

2.1 Practical background

Project owner Torbjørn Aasen has been involved in earlier projects with Western Norway University of Applied Sciences, however the group and project owner has not collaborated in the past. Torbjørn Aasen shared his visions and problem with the group, which in return made it easier to design and develop a solution which can be used by project owner and create value in the future.

2.1.1 Project owner

The project is carried out for Haukeland University Hospital where Torbjørn Aasen is a researcher who belongs to the department of National Competence service for Vestibular Diseases. He specialises in mathematical chaos theory and dynamic systems.

T. Aasen has published over 25 articles[2] on different topics in mathematical chaos theory and its influence on dynamic systems. The last 30 years he has worked on Haukeland in National Competence service for Vestibular Diseases, where he has worked on many different projects, and some collaborating with Western Norway University of Applied Sciences.

2.1.2 Previous work

During the preparation phase of this project the group had a guided tour of the balance laboratories at Haukeland University Hospital. During this tour the group was given an introduction to different software and machinery used to measure balance. The group were shown a demonstration of some software which can calculate area and curve length amongst other parameters, but as Torbjørn Aasen pointed out on the tour, none can display the turning point for maintaining balance. This means that all tests need an expert to give his or hers subjective assessment of the results.

The software being used in these laboratories acted as inspiration to the design of the new software and the group used the tour as inspiration.

The turning point (TP) is defined this way: "*The Hurst exponent is a measure of long-term memory of a time series. The turning point is estimated as the time where the derivative, the stepwise difference of the slope, of the Hurst scaling series crosses from higher to lower than 0.5 (i.e. switching from a positive to negative correlation)."*[3] (Aasen, 2019,p1)

Another definition is that the turning point *"... estimate the time window length [...] during which the regulating functions for maintaining the upright position come into action."(Aasen, 2019, p1).*

Figure 5 shows a screenshot from Aasen's article explaining the Turning Point algorithm in detail.

2 Aasen, T - Information about research, available from: http://aasens.net/research.html (accessed 01.04.20)

3 Aasen, T. Nordahl, SHG. Goplen, FK. Knapstad, MK (2019) The Turning Point for Maintaining Balance, *J Diagn Tech Biomed Anal 8:1.*

The turning-point algorithm

We applied the algorithm to the shift of the Center of Pressure (COP) at the sole of the feet 20 s duration of body sway. The sample frequency was 100 Hz, which gave a time series recording of 2,000 data points.

We have a time series x_i,

$$\{x\}_{i=1}^{n}$$

First, we calculate the mean square displacement $< x^2 >$ in mm for increasing the window size Δt.

$$< \Delta x_i^2 >_k = \left\{ \frac{\sum_{i=1}^{-k+n} (x_{i+k} - x_i)^2}{(-k+n)} \right\}_{k=1}^{\beta}$$

The window size is defined by the sampling frequency of 100 Hz, which sets the resolution to 10 ms. Increasing k increases the window size in steps of 10 ms. n is in our case 2,000 and we increment k until we span a window length of 5 s (β=500), long enough to fulfill the criteria of crossing 0.5 (if a subject does not reach the TP and activate the feedback mechanism to keep the upright position before a time span of 5 s, he/she will certainly fall).

Then, from the scaling properties, we find the slope s_i.

$$s_i = \frac{1}{2} \frac{\log(< \Delta x_i^2 >_k)}{\log(k)}$$

The TP is the time where the derivative s_i', the stepwise difference of the slope, crosses from higher to lower than 0.5

Figure 5 - The turning-point algorithm (Aasen, 2019)

2.1.3 Initial requirements

The group were given some initial requirements for development of the application. These requirements were as followed:

- Be able to gather data from Wii Balance board
- Computing and display of data
  - "The turning-point"
  - Area
  - Curve length
- User should be able to define tests in application
  - Length of test (in seconds)
  - Define what kind of test that should be carried out
  - Let user change between tests when desired
- Exporting of data: raw data and computed data to csv, excel or similar
- Application has to be open-source
- Further development on application should be easy
- The application should be able to run on most operating systems (Windows, Linux, MAC OS)

2.1.4 Initial solution

The initial solution presented to the group was to further develop an already existing open-source application called Romberg Lab (see Figure 3). This application can be found on github. The group discussed the possibility to develop an application from scratch instead, this idea will be further discussed in chapter 3.1. The application is intended to help researchers and doctors alike to gather data from a Wii Balance board to further research on balance.

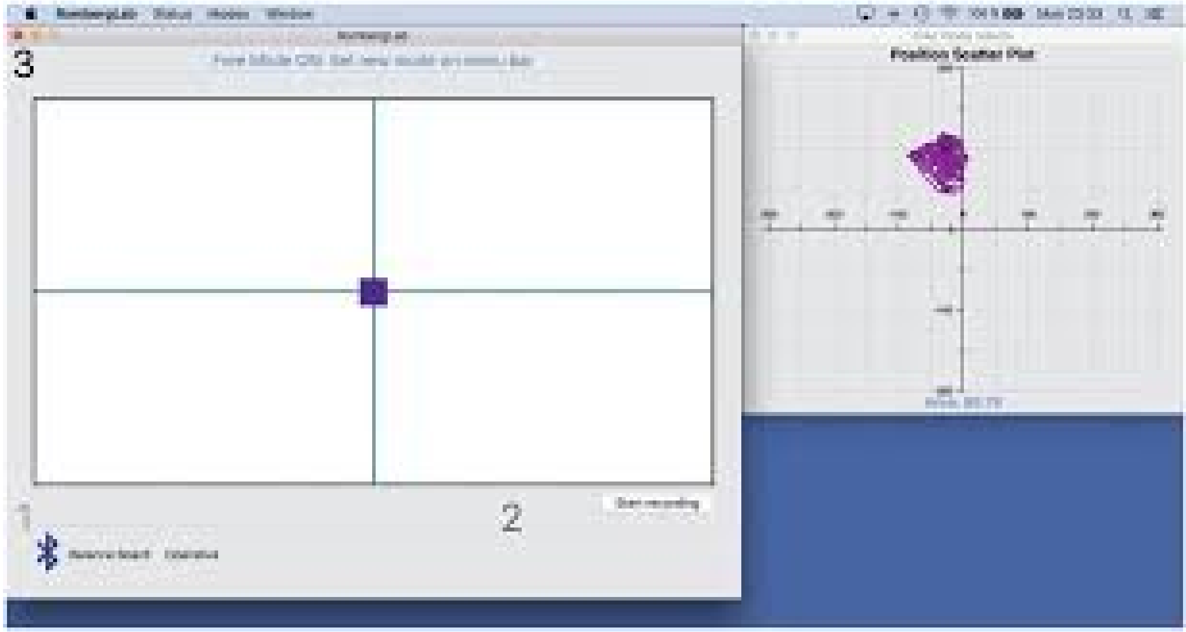

Figure 2.1.4 Screenshot from RombergLab displaying the COP plot (left) and a test result (right)

2.2 Literature background

Torbjørn Aasen made the primary foundation for the project with his article "The Turning Point for Maintaining Balance". The article presents arguments for an objectively measured parameter that can be used for diagnosis of patients through an algorithm that identifies the turning point to maintain balance.

Outside of this article the group has used an article about the precision and accuracy of Wii Balance Board[4]. The conclusion of this article is:

"... the WBB [red. Wii Balance Board] *is an inexpensive, portable device that may be useful for measuring vertical ground-reaction forces and COP with limitations on accuracy and precision. As the WBB is about an order of magnitude less accurate than a laboratory-grade force plates in vertical force measurement, and two orders of magnitude less accurate than recommended for COP location for posturography, it should not be considered to be equivalent to laboratory-grade equipment. However, if calibrated with laboratory-grade force plates, the WBB may provide an estimate of force and COP measures that could be useful for situations where lower accuracy and precision is acceptable."(Bartlett, 2015, p227)*

Bartlett and his team also found these 5 concluding points about the board:

1. *WBB is a low-cost, portable force-plate with trade-offs in measurement uncertainty.*

[4] Bartlett, Harrison L, Ting, Lena H, & Bingham, Jeffrey T. (2014). Accuracy of force and center of pressure measures of the wii balance board. *Gait & posture, 39*(1), 224-228.

2. *Total uncertainty across WBB's is ±9.1 N for force and ±4.1 mm for CoP.*
3. *Repeatability within a single WBB is ±4.5 N for force and ±1.5 mm for CoP.*
4. *Wear does not significantly impact the performance of the WBB.*
5. *WBB factory calibration values are comparable to empirical calibration values. (Bartlett, 2015, p228)*

Another article has been written with the title "Validating and Calibrating the Nintendo Wii Balance Board to Derive Reliable Center of Pressure Measures". The conclusion of this article is very similar and states:

"The WBB [red. Wii Balance Board] *should not be used as a replacement for the "gold standard" laboratory grade force-plate when measuring CoP under both static and dynamic conditions, as it is a uni-axial device and lacks the accuracy recommended for posturography [27]. However, when calibrated with the "gold standard," as done in this study using an AMTI force plate (AFP), the WBB may be used to estimate time-domain CoP measures with improved accuracy."(Leach, 2014)*

This article also includes a lot of results and measurements which can prove valuable to this project. The conclusions of both articles have convinced the team that the Wii Balance Board is the right instrument to use in this project and the group will look into including the findings in the latter article in the project and add a calibration element to the software.

A third article by T. Weaver also had some very positive findings about the Wii Balance Board and mentions:

"Overall, our results generally support the use of the WBB as an alternative to laboratory-grade force plates for center of pressure measurements during quiet stance sway. Therefore, the WBB may be a useful tool for clinical settings where research-grade force plates are not appropriate, or for researchers who wish to conduct field study measures of quiet stance balance control outside of their laboratory."(Weaver, 2017, p54)

This project will use these articles to build an application that can present a usable and reliable way to test, measure and quantify balance with a new parameter, the turning point, which can give a different insight into a person's balance.

3 METHODOLOGY

The project will be completed alongside T. Aasen and M. Kalland Knapstad from Haukeland University Hospital. The group will aim to implement and improve the requirements from the project owner through ongoing conversations and discussions with the project owner. Initial conversations with the project owner will help the group decide the optimal technologies to be used throughout the project.

During the software development phase the group will use SCRUM as a framework. SCRUM is defined and explained in 3.2.

Testing of the software will be done in cooperation with HUS. The product will be tested up against the already existing laboratory equipment. These test will help determine if the application performs as expected and if the outcome is as expected. The turning point parameter will be tested by running the same data through T. Aasen's algorithm and the application and then compare the outcome.

3.1 Preparation

First the group will research previous software used to measure and test balance. This research will help the group get an idea of the desired outcome. This will help decide on the technologies used in the project.

During the preparation phase the group will find articles and sources to back up decisions regarding hardware and software used in the project, in order to catch potential problems early.

Then the group will have discussions with the product owner about the design and specific requirements for the minimum viable product (MVP), which will form the basis of the first iteration of the software development process.

During this phase the group will have conversations with HUS and T. Aasen in order to gain useful insight into requirements and expected outcome in order to choose the best suited technology and be able to not only implement the requirements, but to improve on them.

3.2 Software development

The group intend to develop the software by applying the agile framework SCRUM.

SCRUM is a much used and popular framework based on iterations as opposed to the traditional waterfall model. In SCRUM the essence is that the entire waterfall model is executed in each iteration, thus providing a more agile and flexible development process resulting in a product developed according to the evolving requirements and needs of the product owner. SCRUM ensures a well tested product with desired features.

3.2.1 MVP iteration

When the preparation is over the team will start developing the MVP.

The first iteration has the purpose of creating a minimal viable product which satisfies the minimum requirements defined during the research phase.

Once the mvp is developed it will be tested. The software will be tested with unit tests through the testing framework JUnit. This helps ensure correct implementation of all the algorithms in the software. The testing will also include user tests, mainly from the product owner, to ensure usability and to enhance the user experience.

After testing the mvp the group will evaluate the product with the product owner and generate a list with requirements for the next iteration.

3.2.2 Nth Iteration

The SCRUM process will be repeated as many times as time and resources will allow, but the team has a goal of completing at least two iterations, one MVP and one with additional features and corrections. The reason for the limited iterations is the nature of the project, it spans over only a couple of months, and the resources available to the group makes it hard to accomplish more than two.

3.3 Evaluation and final testing

The software will be tested up against the laboratory equipment at HUS. Similar tests will be run on both platforms and then compared. Since the product implements some of the same parameters as the laboratory equipment, these parameters can be compared and the product can be evaluated.

The group will evaluate the product with HUS and provide documentation for all parties interested.

The product is meant to be open sourced so that anyone interested can continue the development of the product or adjust it to other circumstances or hardware. Therefore it is a high priority to provide sufficient documentation for the application.

4 PROJECT DESIGN

Having an understanding of the multiple solutions to the project is important, and without a doubt there are other ways to solve this project. In this chapter the possible solutions are presented, together with tools, risks and development method.

4.1 Possible approaches

The project owner’s suggested solution is explained in section 4.1.1, however it was not a requirement to continue developing with this solution. The main requirement behind the project was that the solution should be able to use the Wii balance board as a platform to conduct tests with and show the same results as the more expensive tools they use. With this in mind the team could look for multiple solutions and develop further with the most optimal solution.

4.1.1 Further develop open-source application in Object-C

This initial solution was presented by the project owner in a lecture. The solution was built upon the idea of using the advantages of open-source to further develop an already existing open-source application called "Romberg Lab"[5]. This application already gathers data from a Wii Balance board and displays information back to the user, however this data was irrelevant and not usable for the project owner. Romberg Lab was also written in Object-C and compatible with MAC OS exclusively, which left a lot to be desired, specifically being able to run the application on multiple operating systems. The team discussed the possibility of developing this application in a docker container, meaning the application would be able to run on multiple operating systems.

In theory this was a good solution, however as the group needed good insight in the code written and spend time learning an unknown language. The group decided that learning both the foundation of the code, which at times were a mess, and a new language would imply not being able to keep up with deadlines. The application is also quite outdated, meaning the group would have to make substantial changes to the application as well. Following these arguments the solution under was suggested.

4.1.2 Develop own desktop application in Java

Following the solution above the group found a java library that gathers data from a Wii Balance board comparable to Romberg Lab. The team decided to discuss the possibility of developing a desktop application with the project owner. This way the team could develop an application tailor made to meet the requirements of the project owner. It would also be able to run on multiple operating systems, since java is more or less universal.

The application will be developed in Intellij and written in java, which both team members are comfortable with. The team will also use javaFX to create a functioning User Interface.

4.1.3 Discussion of alternative approaches

The ideal solution is to gather data from an arbitrary Wii board, display information dynamically while a test is being conducted and upon completion of such a test display important information such as; curve length, area and "turning point". The Project owner would also prefer if the application supports multiple operating systems and not locked to a singular system like MAC OS.

The team feels they hit the mark by choosing to develop their own desktop application in java. Even though both team members are comfortable with java, there are still some new-found elements in javaFX that need to be learned. With respect to deadlines the team are confident in their abilities to deliver, and believe nothing will be overdue.

---

[5] bendermh, *RombergLab.* Accessible from: https://github.com/bendermh/RombergLab (Accessed: 12/02/2020)

4.2 Specification

The team chose to develop their own desktop application using Intellij in combination with java and javaFX. Both members prefer to use Linux as operating system on their personal machines, therefore developing further on Romberg Lab seemed less likely as it is locked to MAC OS. Even though a potential docker solution could have potentially solved this, the group members are both more comfortable with java and making a tailor made application for multiple operating systems seemed more likely using java. Developing a desktop application would also fit the project owners needs and requirements better as it can be custom made.

4.3 Tools and programming language

4.3.1 IDE - Intellij

The team prefers to develop in Jetbrains' Integrated Development Environment (IDE) Intellij. With embedded support for database tools, git version control and maven, Intellij wants to make sure productivity stays at a high level for developers all over the world. The team also prefers the layout in Intellij and both can navigate fast and efficient through the project.

Choosing an IDE first and foremost comes down to preference. The group has used most of the study developing in Intellij and both prefer the features Intellij has over other similar IDE like Eclipse.

4.3.2 Maven

Apache maven is a tool that specializes in project organization for java. Maven collects all information with regards to the project through a project object model (POM) file. In this file there exists a dependencies list, which includes all dependencies, libraries and packages the project relies on. This way a uniform build system is created and makes it easy to start up new projects without having to import everything individually.

The group relies on Maven to set up the development environment where the group will work in. Both group members take advantage of Maven making sure the environment is the same every time the project is launched. Maven also automatically updates the project if the POM file is changed.

4.3.3 Version Control - Github

Github is a web based hosting service for version control using git. The group is able to make a private repository where the project will be uploaded and run version control. Through github the group can also make the project open-source upon completion of the project. Taking full advantage of github the group works in git branches, where one branch is assigned to one feature. Using branches is efficient and minimizes potential merge conflicts that can happen.

There exist similar alternatives, more specifically GitLab, but the team will struggle taking full advantage of GitLab's self-hosting feature which makes GitLab more desirable for a bigger team. The team members are also both familiar with Github and its features.

4.3.4 Programming language - Java

As the group has chosen to work in JetBrains' Intellij the prefered language is java, Intellij is designed as an IDE for java development after all. Early in the startup phase of the project, there were discussions about other alternatives, however the library gathering data from the Wii board is written in java which made other alternatives less desirable. The library will be discussed in further detail below. As mentioned earlier the group is comfortable with java and is compatible with most operating systems.

4.3.5 Libraries / API

JavaFX

JavaFX is a development platform where developers can develop desktop applications compatible with most systems. JavaFX uses the concept Model-view-controller (MVC), which both team members are well versed in. Developing an application using MVC to live up to the requirements given to the group should not be a problem. JavaFX also provides features to create a user interface through scene builder and connecting these interfaces to controllers. Scene builder works together with javaFX to create a wall between the components in the application (UI and logic). Scene builder utilizes a drag and drop UI to create the UI of the application with ease. These "scenes" are fxml files and are being displayed on screen as UI.

Bluecove

BlueCove is a library which provides a necessary implementation of JSR-82, a way to use bluetooth in the Java language. With BlueCove the application will work on multiple bluetooth drivers and should work across different operating systems, which makes it very attractive.

Wii Board library

The project will use an already implemented library to interact with the Wii Balance Board. The library is open source and originally developed by the Github user *pierriko.* This library has been used in other projects including the Wii Balance Board and the library has good documentation.

4.4 Project development method

4.4.1 Development method

The team has chosen to use the agile development method scrum and already early in the project phase decided to work iteratively with 1 week sprints. Every week the group will discuss project status, work done and plan ahead for next week. The team will also update their blog with a summary of the week.

The team also uses a trello board (todo-doing-done) to organize tasks at hand. Through this board the team can distribute tasks between both team members, as well as keeping track of finished and unfinished tasks.

Following up on trello board, Github branches are implemented for each feature the team wants to implement. This makes it easier for both team members to work on their given tasks without interrupting the other on their branch. The master branch is also off limits and only the last runnable version of the application is run there. These are good programming habits which the team wants to have in general.

4.4.2 Risk management

With every project comes risk and the group has mapped out the most prominent risks in a risk analysis. Look up appendix A for further information regarding risk.

4.4.3 Project plan

The project has been mapped out in a GANTT chart to show deadlines and time constraints the group has to maintain. Look up appendix B for further information regarding project plan.

4.5 Evaluation method

The evaluation of the project will be done in cooperation with the product owner. Since the project uses the SCRUM framework there will be some evaluation after each iteration. This will help ensure good quality on the product.

The team will use unit testing to test the software units and ensure quality in the code. This will also help the team find errors early and help develop a good product.

The team also plans to test the product up against laboratory equipment to test and compare the product with the lab standards.

Testing of the UI will be done in collaboration with the product owner. Since the product owner is no IT expert nor interested in an over-complicated UI, the group has decided to develop a UI which will be natural to the product owner and test it with the product owner.

The product will also be tested on different operating systems to ensure stability and define how to best use the product. If any issues should arise on different operating systems, the group can document this and help the end user setup the environment accordingly.

The main areas to evaluate are:

- The product is working
- It performs well compared to existing (and expensive) technology
- The user interface works and can be understood
- Researchers know how to use the product.

5 DETAILED DESIGN

There is a lot to consider when developing a desktop application from the ground up. What components are needed to make the application work? Do every component need to be tied to a purpose? How can this be handled? How to define a suitable User Interface for the product?

This chapter discussed architecture, implementation of key features and UI design choices.

5.1 Architecture

The application revolves around multiple components to make it function as a whole, with this fact in mind selecting the right architecture is important. Which tools and libraries are also important, however these were discussed in chapter 3 and were chosen before developing the application.

5.1.1 Exporting data

Considering that the group was tasked with making a cost efficient solution and that csv files are so lightweight to store, choosing to store the completed tests on the machine was the most efficient solution. Storing the completed tests on the machine was also a discussion earlier in the project when the group and project owner discussed Romberg Lab, as its solution to storage is also to store files on the machine. One could make the argument that online storage is superior, however relative to the project it would be overkill to implement a cloud solution with a corresponding storage plan. As the application is supposed to be cheap and open-source, online storage would never have worked in the first place.

As previously stated, the application will be exporting the completed tests as csv files to the machines' filesystem through a directory chooser. Using a directory chooser the application prompts the user for a location on the machine where the csv file will end up being stored, it does this through the use of an absolute path to a directory the user chooses. The file saved will also contain the name of the person conducting the test, which test is being conducted, date and time for the user's convenience as shown on Figure 5.1.1.

	A	B	C	D	E	
1	Id	David				
2	Test type	Romberg				
3	Duration	20				
4						
5	Test	1	2	3	4	
6	TP	2.2693824701195253	0.41374344718030187	4.193123809523819	1.6762294303797434	
7	Area	748.040457530171	210.67440351630967	931.0255977259221	1277.5090241928017	
8	Curvelength	495.42677178201086	301.9089914025956	487.0566979035116	431.08015671444	
9	Curvelength X	197.26793776625928	165.95771493483227	273.41807401217017	202.56182912977542	
10	Curvelength Y	404.2851221888884	209.10591649149802	338.16698102282936	328.63057543247584	
11	Time 1	X 1	Y 1	Time 2	X 2	Y 2
12	0.007	12.910152925530973	58.41578795827735	0.017	-9.107472418340361	2.4
13	0.017	13.448924327848125	59.293554475237414	0.028	-8.686165904163907	2.4
14	0.04	14.160756227717364	60.06114769165917	0.047	-8.237170675149272	2.0
15	0.048	14.916629399344053	60.656777746146645	0.067	-7.658205760118821	1.6
16	0.068	15.224605225580257	60.53621406786307	0.077	-6.927416669795718	0.6
17	0.091	15.56541565073071	59.94766282574372	0.098	-6.469047758452652	0.3

Figure 5.1.1 - Screenshot of exported CSV file with test data.

Writing files

Upon completion of a conducted test or multiple tests the user can choose to export these conducted tests if desired. By pressing the “save” buttons the application calls csvwriter to write all computations and raw data to a csv file. The csv file contains turning point, area, curve lengths and all gathered data from the wii board since the start of the test.

Properly exporting data

It is important to be able to recognize and make it convenient to find the exported csv file for future reference. With this in mind the team added a couple of checks before exporting data. This includes prompts on screen if ID field is empty or test type not defined as it is crucial to classify who conducted the test and how the test was completed for future comparison. Furthermore the application is going to be used for research and it is important to be able to locate previous tests with ease.

5.1.2 Application architecture

Considering there is a lot of dynamic data that is being presented to the user, the application needs to have a robust model behind the surface. The application is built on the capabilities of the Wii board to do accurate measurement, and therefore the model needs to be able to handle dynamic data.

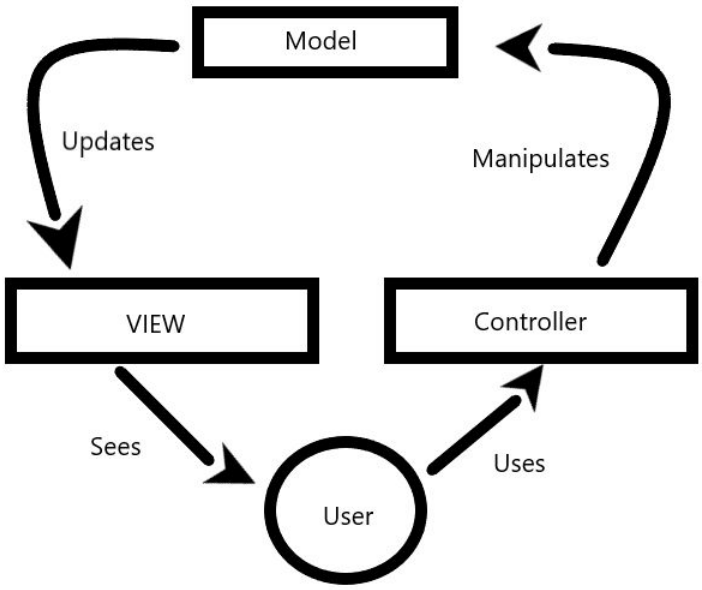

Figure 5.1.2 Model-View-Controller (MVC)

The primary model in discussion was Model-View-Controller (MVC). MVC is one of the older design patterns, but also one of the more robust. It is popular in web development, but traditionally used for desktop applications MVC separates the responsibilities in their own area. Looking at figure 5.2, the full scope of MVC is portrayed, starting with user and ending with view shown back to user. This way when the user is testing balance with the Wii board the application can update the view using the dynamic data being gathered from the board.

Controller

The controller acts as a link between the model and the data source, the Wii board. It accepts inputs from a user that is currently conducting a test on the Wii board and further passes these inputs to the model or view as commands. As seen in figure 5.2 the user is directly responsible for what the controller does, before it can further manipulate, compute and display data. The controller keeps view and model separate. This application relies on the controller to set up the dashboard and function as a link between the model that handles the background logic and the view that displays the test being conducted.

Model

In this specific design pattern the model acts as the central component. The model is independent and does not rely on the user interface, it acts as the application's dynamic data structure and acts as a back-end where it can manage the logic of the application. The application is structured in a way that it can take advantage of this concept, where the model is being supplied with data from the controller and can further act upon this data.

View

The view is displayed back to the user and is a visual representation of the work done by model and controller. Both controller and model can change how the user looks at the view, be it data in forms of charts, plots or tables. It can also be pure numbers being displayed. The application handles the view in such a way that data from the initial testing with Wii board becomes plots and numbers displayed in their respective fields. Initially stage setup is in controller and a FXML file is the scene displayed, this will be discussed thoroughly in chapter 5.4: UI design.

Model choice discussion

MVC is well known to both group members and they have used it on multiple occasions during their studies. While both being familiar to the group and a good fit for this application it becomes clear that MVC was the model to be used. The application relies on its main components Wii board, logic and GUI. These components serve their own purposes and follow the MVC principle without doubt. The report will go more in depth about these main components below in chapter 5.2.

5.2 Application components

When developing this application, multiple components were needed to develop a usable, functioning and accessible application that could further be used for research. These components consist of Wii board (gathering of data), logic and GUI as seen in figure 5.2 below. The components are designed so that they are independent and can be replaced if necessary; e.g replacing Wii board for another source of data gathering.

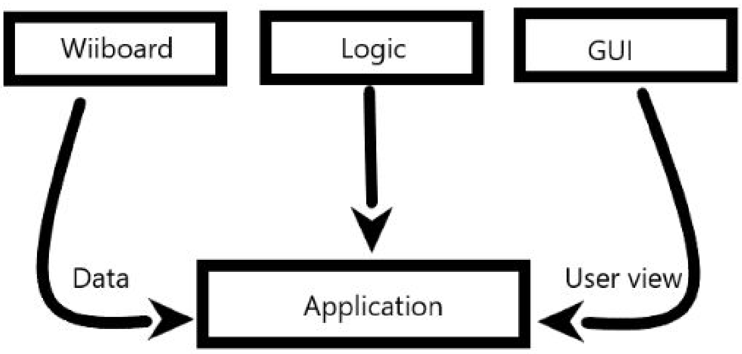

Figure 5.2.1 Components in this application.

GUI (Graphical User Interface)

Considering the application is developed using javaFX, the GUI will function as the main component for the application. The controller can be found under the GUI, together with the main() method, where the application is launched. The gui is directly responsible for the user's view and updates information dynamically, records data, cleaning up excess data after a completed test, exporting data; where the user is directly in control of what is happening. The user can navigate the application and choose what to do at any given time if the option is available, as an example: the user can't export data if no test has been conducted, however the user can navigate the empty plots and prepare a new test if so desired.

Logic

This component consists mostly of computations and algorithms, however the file manager also resides in this component. The main purpose of this component is to produce results from the tests being conducted through the Wii Balance Board. During tests the COP measurements will be updated live, however TP, curve length and area are withheld until completion of the current test. Upon completion of a conducted test the data gathered will be run through the algorithms and displayed on screen. There is also implemented an interface for the logic component, making it accessible and easy to use when further developing.

The file manager is only a small part of the logic component. It provides the rest of the application with a way of writing CSV files and finding directories on the user's system. Further details regarding CSV writing and exporting of data can be found in chapter 5.1.1.

Wii board (data source)

The Wii board is considered a crucial component, because it works as the data source for the application. Without it the application will function and the user can navigate the view, however since no data is being fed to the controller the application is more or less obsolete. The application as a whole relies on the Wii board to produce data while a user is undergoing a balance test. During tests the application is displaying a live feed of the test and the user's balance, however statistics is displayed upon completion of the test. These statistics include TP, curve length and area. As shown in figure 5.2.1 the Wii board component is directly responsible for the data being supplied to the application, however it does not hold computations as mentioned under logic.

5.2.1 Component dependencies

As discussed in chapter 5.2 the main components are independent, however they still reference each other frequently to update the application as needed. Considering there is a lot of data gathered, computed and displayed these references are necessary to develop a fluid and responsive application. This concept is best illustrated as a triangle, as seen below in figure 5.2.2. The application relies on each main component to fulfill their respective purpose so that it can function as designed.

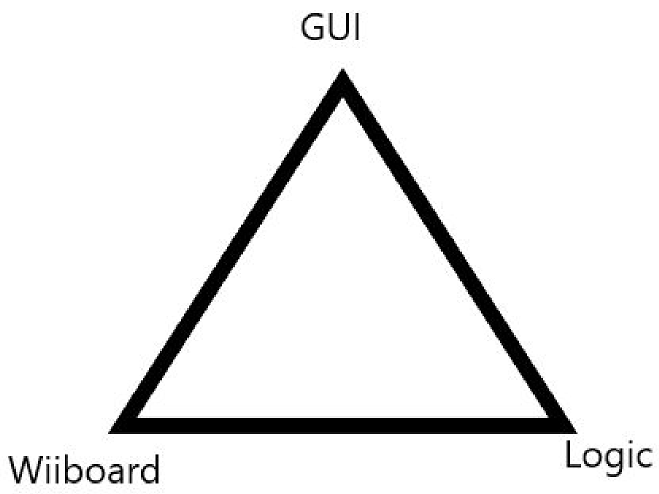

Figure 5.2.2 All components reference each other.

Further discussing figure 5.2.2, the GUI is dependent on updates from both logic and Wii board considering it wants to show the dynamic data at any given time. The logic component is being supplied by data from the Wii board data source and computes the data so that GUI can further display the computed data.

5.3 User Interface Design

When designing the UI the team wanted to create an UI with focus on responsiveness, accessibility and usability. Whether the application is being used on a small laptop, a desktop with a bigger display or windowed mode the UI should still function the same way and provide the necessary information and functionality to make the application usable.

5.3.1 JavaFX UI components

JavaFX provides a simple, yet effective, UI building tool in scene builder. The team made a draft of the UI to start using the application early on in development. This made a huge difference, considering the team was reading information out of the console instead of any sort of UI.

JavaFX has many UI components, including UI controls, layouts, css, plots and many more to customize and design UI to fit the developers needs (Tutorialspoint JavaFX, 2020). UI controls consist of buttons, text fields and labels, these controls are used to update the application through backend logic. Layouts consist of predefined panes such as Anchor pane, VBox and HBox. These panes can be explained as containers for other components for the UI. All of these UI components and elements are defined through scene builder and written as a FXML file, while using scene builder the team can preview the UI and make changes whenever necessary. Having this visual aspect while developing is also needed to develop the frontend through modifying the css file.

5.3.2 UI Layout

The team decided one "scene" was enough for the application, however since there are multiple plots, statistics and information being displayed there had to be a way of navigating the application. Using the predefined VBox pane the team could create a navigation menu with multiple buttons that cycle through the information displayed to the user as a "view" menu. This concept and the entirety of the application is shown in figure 5.3.2 below.

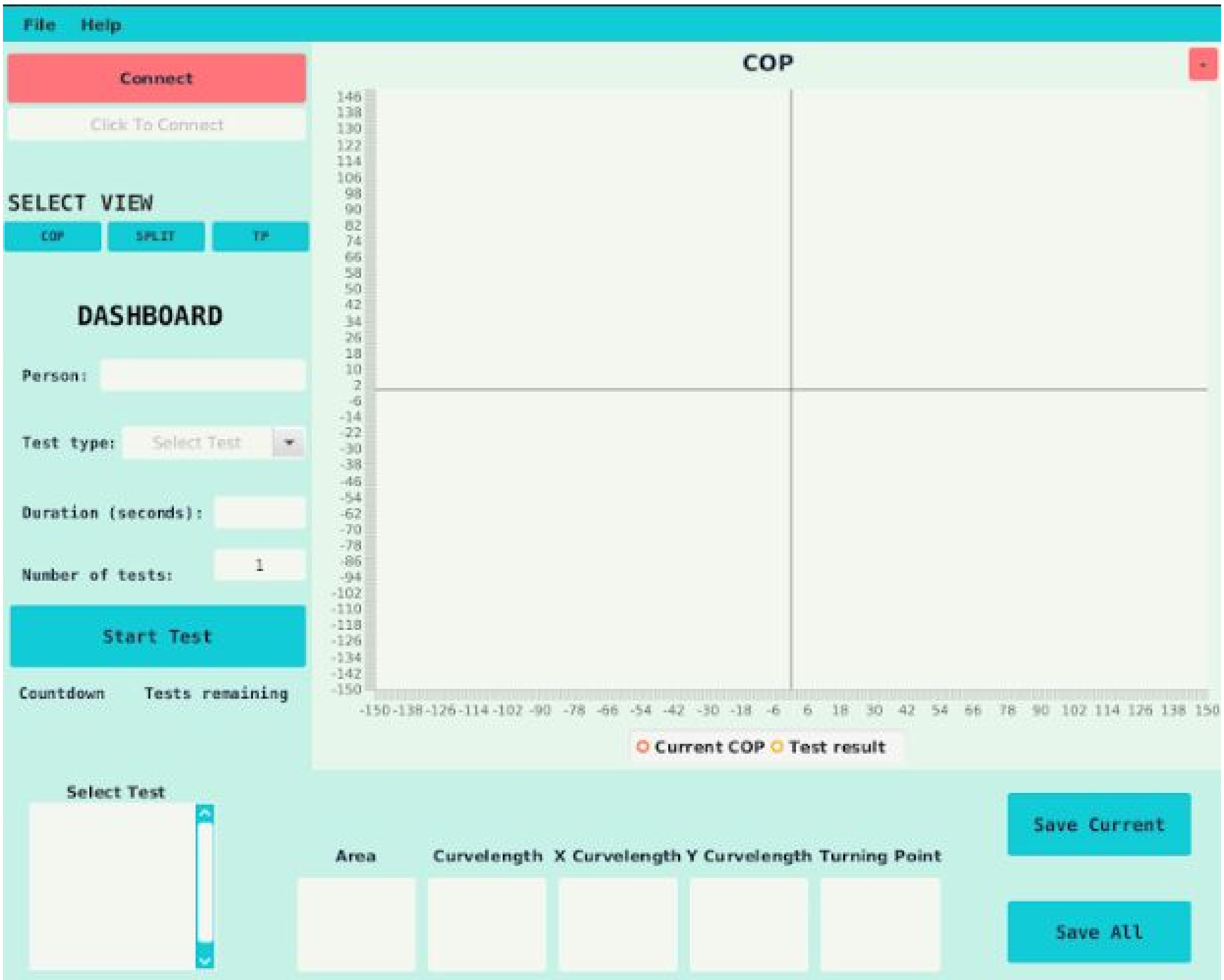

Figure 5.3.2 User Interface for application, see the following chapters 5.4.3 - 5.4.6 for a more in depth explanation of the UI

5.3.3 Connection prompt

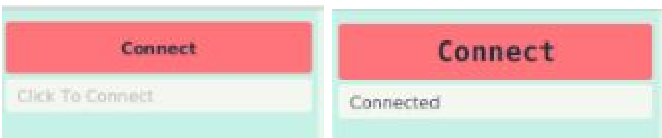

Figure 5.3.3 Connection pane when not connected (left). Figure 5.4.3 Connection pane when connected (right)

This section of the UI gives the user access to connect to a Wii Balance Board through bluetooth. It displays if the Wii Balance Board is connected or not. Pressing the "connect" button gives a prompt to the user to sync the Wii Balance Board with the computer to allow further use of the application as it requires a data source to produce results.

5.3.4 Test menu

Figure 5.3.4 Test menu

The test menu is designed to be accessible and easy to use. It features 4 parameters, as seen in figure 5.3.4, in the form of who is conducting the test, what type of test, duration and how many times the test will be conducted (e.g 2x20 seconds). This menu also has a button of starting the test that the user wants to complete. There is also a timer counting down the duration to let the user know when the test is completed. When the countdown number is black it indicates the remainder of a running test. If it is red it counts down to the starting of a new test.

Choosing a test type is quite simple and it is done by clicking on the drop down menu next to "test type" shown in figure 5.3.4. The drop down menu consists of pre designed tests with the option of adding new tests when desired. Both person ID and test type can be anything the user wishes.

5.3.5 Plots and statistics

A B

Figure 5.3.5 Center of pressure (A) and X-Y split (B)

Being a very visual application the plots and statistics are important to gather necessary information on a test. Choosing what plot is displayed can be changed at any given time through the buttons under the “Select view” menu in figure 5.3.2 (COP, X-Y split, TP). These buttons can be used to change plots displayed during a test or after the test has been completed. COP and X-Y split can be seen above in the figures above, while the different plots regarding TP can be seen below. If the plots are too small to analyze it is possible to zoom in on the plots as desired.

Figure 5.3.5 The turning point visualization

Here the top left (Xi) is the time series calculated by integrating the COP points. This plot is the foundation for calculating the other plots and parameters on this page.
The bottom left plot (MSD) is the mean squared displacement plot, which says something about the displacement and the movement of the patient.
The top right plot is the same plot (MSD) but on a double logarithmic axis.

Finally the bottom right plot is the Turning Point plot. The turning point is where the curve goes below 0.5 on the y-axis. In this example it is 1,68 as seen on Figure 5.3.6.

5.3.6 Result

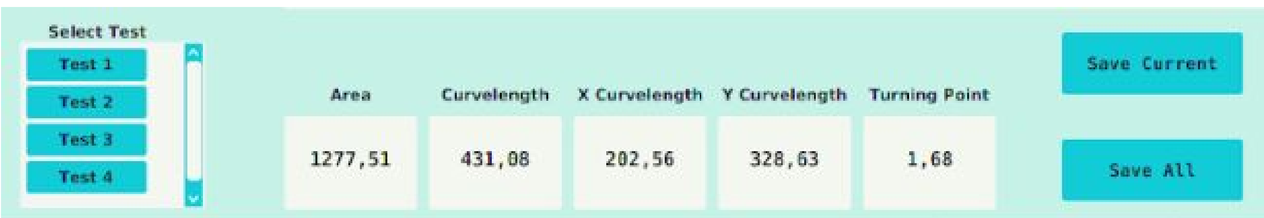

Figure 5.3.6 Result information displayed together with “save” buttons and conducted tests

Looking at the result section of the layout the user can see how they held up with respect to area, curve length and TP. This section requires more knowledge on balance in order to be able to use the application efficiently, but as this application is first and foremost designed for research on balance it should not be a problem to the main user group. If the application is to be used by none-researchers the parameters will require more explanation and research.
If the user has run multiple tests, these tests will be shown as in figure 5.3.6 and it is possible to further go back and look at each individual test by choosing e.g test 3. There are also buttons to save the current test or save all conducted tests to the machine. This allows for further analysis and comparison between future tests.

5.4 Use Case

Considering this application is mostly about visualization and computations than anything, determining clear purposes for the application can be difficult. However, as discussed in earlier chapters, the application is meant to be used for further research on balance and since the application is open-source further development is bound to happen in the future. As it stands now there is only one use case for the application.

5.4.1 Primary use case: Start and conduct a single test

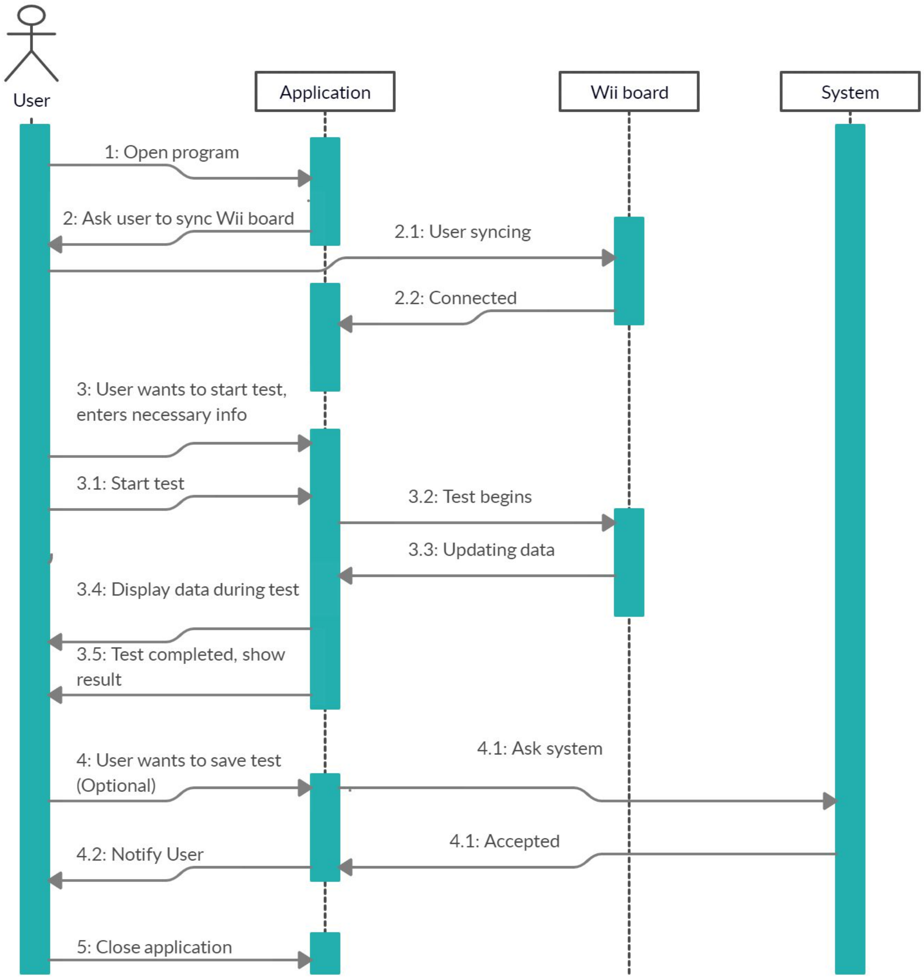

Figure 5.4.1 Sequence diagram for simple use case of the application

The user opens the application with testing and analysis in mind, however before a test can begin the application has to connect to a Wii balance board to function properly. As shown in figure 5.4.1 the test does not officially start before 3.1. The user has to make sure the Wii balance board is connected, once connected there is a "connected" prompt displayed on screen to let the user know. Having connected the Wii balance board the application is ready to run tests (Illustrated in figure 5.4.1, step 3). The user enters necessary information regarding the test, which are being used as input parameters for the application. These parameters contain person ID, test type, duration and how many times the test is being run. Once the test is being conducted the Wii balance board supplies the application with data, and this data is being further displayed on screen. Upon completion of the test, the results are shown back to the user and it is possible to store the completed test. Storing the test is optional, because there are cases where the user might not stand properly or conduct a test incorrectly, making the test obsolete.

5.5 Deployment

The application is exported as an executable jar file and supports java 8 and newer versions. Having the file accessible for future downloads is important and the file will therefore be available on the project github page. The github will be made public upon completion of the project and after the final deadline.

The team packed the application using a plugin for Maven that assembles jar files. This assembly plugin packs all dependencies and the application as a whole, and produces this executable jar file.

Following deployment good code documentation is needed. The application is open-source so future developers can improve or add features upon the application at will, and if no good documentation is present other developers might be confused when looking at the team's code. The team members certainly felt this way looking over the RombergLab code. The documentation describes how functions work, explain certain quirks in the application and should make the code behind the application easy to understand.

6 EVALUATION

6.1 Evaluation method

The team and the project owner both agreed on performing evaluation after the first prototype, considering evaluating the application when it is half way done without the main components would be inefficient. The evaluation, although not many, would be more useful to develop a good mutual understanding for further improvements upon the application. UI, visualization of data and the algorithms behind the computations are all topics for the evaluation of the first prototype.

Having said this however, discussing the need to make the application easy to use is important. The main user base is first and foremost researchers and doctors that want to further research on the cheap alternative with Wii balance board. They have limited experience in IT and making the primary use case easy to understand is important. Evaluating the applications' effect on balance research and how it will impact future balance testing is not something the team can partake in as research has not been conducted yet.

6.1.1 Technology Readiness Level

Phase	TRL	Hardware	Software
Research	1	Basic principles	
	2	Concept and application formulation	
	3	Concept validation	
Development	4	Experimental pilot	
	5	Demonstration pilot	
	6	Industrial pilot	
Deployment	7	First implementation	Industrialization detailed scope
	8	A few records of implementation	Release version
	9	Extensive implementation	

Figure 6.1.1 Technology Readiness Level (Available from: https://www.researchgate.net/figure/Technology-Readiness-Level-TRL-scale_fig1_321013240 accessed 29.05.20)

Mentioning the Technology Readiness Level (See figure 6.1.1) is important when discussing the evaluation method. Considering the team suggested to build an application from scratch some basic principles had to be covered, these included the questions; “Will it work?” and “How will it be done?”. The team found a library and answered these questions by producing a very first concept that could gather data from the Wii balance board and produce results in a console. This would cover TRD 1 and make a solid foundation to further build an application that would suit the project owner’s needs.

Further discussing TRD 1 - 3 the team also had to prove the application would produce the same results for TP as the project owner. If the team managed to produce the same results it would further ensure that developing the application would become a reality. The team managed to provide the project owner with results that were up to par with the original results for TP, meaning the algorithm was implemented correctly. Having the same results also proved that the Wii-based solution would work just as good as T. Aasen’s own solution (See figure 6.2).

Further proving that the Wii-based solution would work the team needed to give a very first prototype that could run the primary use case in full. This falls under TRD 4 - 5 and would become the prototype the team would further improve upon until release. While being back and forth with the project owner after the initial demonstration of this prototype, the team managed to provide the additional features and cleaning up the functionality to further climb the TRD scale up to level 6.

The weeks before the final deadline the team managed to achieve the deployment level of the TRD scale and it was possible to use for extensive testing if desired. This very first implementation would also be the team's last version and would be labeled officially as the release of the application. On the TRD scale this would classify as level 7.

6.1.2 Testing and User Testing

Devising a plan to undergo thorough testing is needed to secure the functionality and usability for the application as a whole. Originally the team had planned to test the Wii board solution directly up towards Haukeland's own machines with regards to performance, deviation and accuracy. Due to the global Covid-19 pandemic, however, the team has been forced to improvise and change their method of testing, especially with regards to user testing and the machines on Haukeland.

With respect to the global effort put into combating Covid-19, the team has put more emphasis on OS testing, unit testing and ensuring the application is functioning at an optimal level.

User testing has been strictly limited to the team members itself, while not being ideal it has helped a great amount in calibrating and developing the application to be more user friendly. Not having a larger test group is also limiting the team to getting more feedback, and to combat this the team has sent multiple demos to the project owner for additional feedback.

6.2 Evaluation Results

As shown in Figure 6.2 we see that the results achieved by the group are exactly the same as those of T. Aasen on the same dataset. This shows that the turning-point algorithm is correctly implemented in the program.

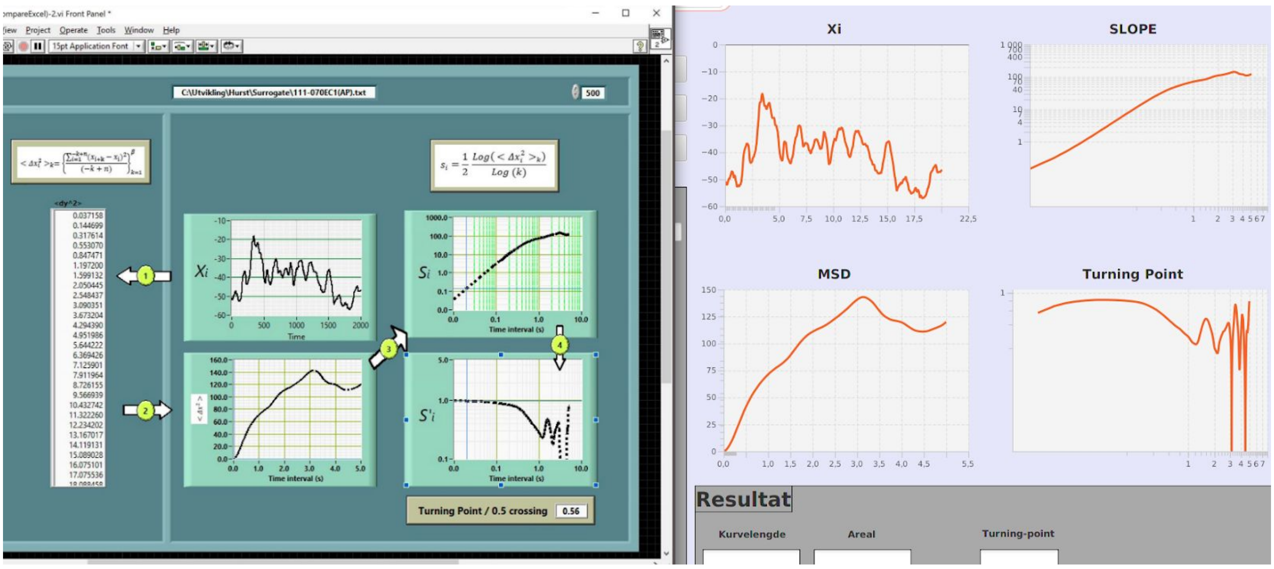

Figure 6.2 Comparisons between T. Aasen's results (left) and the product developed by the team (right).

The application works as expected and produces good results based on the tests the group has been able to perform. The limiting circumstances with the Covid-19 pandemic and the lockdown during developing the product does however mean that further testing could be required.

7 RESULTS

After having developed the application and done extensive testing with respect to functionality, operating systems, computations and user interface, a presentation of results are in order. As expected, the application runs flawlessly on both team members' computers, where both run linux. The unexpected may come when dealing with newer hardware, drivers and operating systems.

The Wii balance board is more than a decade old and was first released in 2007 and the BlueCove library that connects to the Wii board only supports 32 bit systems. Nowadays most systems and computers run 64 bit and are unable to support this service. Newer operating systems, such as windows 10, 8 and newer iterations of MAC OS, also don't support the older drivers which the Wii board wants to communicate with, this was experienced first hand by the team members. However, earlier windows and mac iterations such as windows vista and xp should still be able to use these drivers. These limitations come from technology being too far apart.

To combat this unexpected result the team has decided to recommend the user to set up a linux environment either in a virtual machine or as an operating system that mimics the team members own environment. There is also a chance that buying an usb dongle will let the Wii board connect to the dongle, however this was not tested as the team members did not have such a dongle, but in theory it should work.

In conclusion the result is as expected with respect to the goals and problem definition. The application fulfills the project owner's visions and will be used to run tests on balance with a cheap solution upon completion of the project. The application will be downloadable as an executable jar file that supports versions of java 8 and newer. Therefore, further development will also be possible by the interested people.

8 DISCUSSION

This chapter discussed the consequences of the chosen approaches for developing the solution. How these choices influenced the result will also be covered. Another important question being answered is "how could the result be improved?".

8.1 Importance of a cheap solution

Since the first presentation for this project, where T. Aasen presented the project as a whole, the project stuck out for both group members. Developing a cheap solution that works in a similar way compared to the expensive machines on Haukeland. The concept of making a cheap solution to an already available product is interesting, but could also potentially help a multitude of people further down the road.

T. Aasen has discussed a lot of using the finished product for further research on balance and cost efficient solutions. This project was inspired from earlier research with the Wii balance board in combination with the newly created "turning point" algorithm. Therefore having a solution that both fit the budget constraints and fulfill the necessary requirements for further research is important. The solution the team presented was an application that is easy to use, accessible, open-source and requires only a Wii board and a compatible system (recall chapter 7 32 bit vs 64 bit systems).

The solution is expected to be used by a variety of researchers and doctors on Haukeland. T. Aasen is also very optimistic with this project and believes it is even possible for potential patients to test themselves at home. This concludes the question, "For who is it important that the solution is cheap?", and the answer is simple. Having a solution that is not necessarily better than already established systems, but rather up to par will make everyone benefit, especially when the price difference is massive (Approx 250 nok vs 250.000 nok).

8.2 Choices

The team has without a doubt made choices and that impact the solution presented in multiple ways. Although the choices have impacted the solution, the choices never went in conflict with the initial requirements for the final solution and were helpful to produce a result for both the project owner and the group to be satisfied with.

Early on in the startup phase for the project there was a lot of discussion around the application. Multiple solutions were thrown back and forth, from designing an application in C# with .netCore (this solution didn't make it past the brainstorming as the libraries would not support this route), developing further on RombergLab to the solution that the team went with: developing an application from scratch with Java. Java was first and foremost the frontrunner for all discussion as it is the group's preferred language, but also the flexibility it allows with running it on multiple systems. Having the option to boot up the application anywhere proved to be quite desired. Thinking back to chapter 7 however, the results were not quite as intended and led to the application only working on 32 bit systems, linux and in theory Mac OS. This is not a direct influence from the team's choices, since the technology behind it all, the Wii balance board, is from 2007 and supports old drivers.

Developing the application from scratch also allowed the team to develop a more tailor made application that would suit all of the project owner's needs and visions. The application exceeded expectations and eventually more features were requested during the project development phase. This would not have been possible choosing to develop further on RombergLab as the foundation behind the application is quite lackluster, and thus allowing the group to develop a tailor made application of high degree.

User interface has been quite easy to develop through the use of scene builder. The obstacles regarding UI has been to "beautify" the clean look that the team wanted to provide. The design choices that went into the UI are simple, yet effective, and led the team to the easy to understand and functioning UI the application has today (Recall back to 5.3). While future versions of the application could change the UI slightly or even do a complete overhaul, the UI serves as the view of the application and it is more than enough for the current version.

The project result has been influenced a little by the current global pandemic Covid-19 and as a result the team could not conduct testing up towards Haukeland's own internal machines. These tests would further see how the Wii board solution would fare up against the already established closed systems. Although the closed systems would not provide TP as a statistic, the precision and accuracy would have been interesting to put up against each other.

8.3 Improvements

The team has, without a doubt, developed, designed and presented a functioning, accessible and usable solution that will be used further together with the Wii balance board. The solution also runs perfectly fine on most systems (it is expected that the systems are 32-bit or with the likes of Linux). Discussing how to improve the results however, is a little bit different from most projects. This is because the bluetooth technology in the Wii Balance Board is quite outdated and therefor a limiting factor to the result any developer can achieve.

In order to improve the current result the team would need extensive knowledge about bluetooth and the older technology in the Wii Balance Board. While the libraries are quite easy to implement and develop with, bluetooth is a very difficult technology to use (this was experienced first hand by the team). Considering this flaw the application still runs smooth, is lightweight and can be used by anyone as long as they e.g install Linux, a free and open-source operating system. The application is also open-source and other developers with more expertise on this matter can improve on the application further.

8.4 Future use and further development

The application has a great potential to be a good tool in diagnostics and self testing. When more research has been made on the “turning-point” one would potentially be able to catch symptoms early and use the program to test students, kids and anyone interested in their balance. For example if a patient suffers from dizziness the application could help the patient keep track of the situation with home testing. These tests could help the patient and the patient’s doctors track the diseases’ development.

The application could also open up for studies on balance outside the premises of a hospital laboratory. An example could be a study at a bar on how alcohol influences balance or a study on how kids’ balance change during a school day. One could study if there is a connection between bad balance and bad concentration and so on. All these ideas could be possible because the application is a cheap, reliable and open-sourced solution with a good and proven track record.

The potential usage of this application is huge, however it may require further studies on the turning point and further development on the platform.

9 CONCLUSION AND FURTHER WORK

The team set out to reach the goal of creating an application together with Haukeland University Hospital. The application should be able to take in data from a Wii Balance Board, visualize parameters such as area, curve length and turning point and finally it should be able to export data. A goal was to make a cheap, available and reliable product as opposed to the big, difficult to transport and expensive machines you would find in a balance laboratory.

The main motivation behind this project was a new turning-point algorithm presented by T. Aasen, which would make it easier to make a more objective assessment of a patient’s balance, and this new research resulted in the desire of software that can be used to further study the turning-point algorithm.

The team developed a prototype and tested its functionality and usability, and it was deemed a success. After the evaluation of the prototype the group set out to complete another iteration of product development where new features were added in order to optimize the application. The final product works well and fulfills the initial goals because it is a cheap, reliable and accessible.

Small challenges appeared in the final testing phase where the application was tested on different operating systems. Here the team found that the bluetooth used in the Wii Balance Board (from 2007) did have some limitations on the newer 64-bit operating systems. These challenges are minor since the application works well on Linux, a free and open-source operating system. This means that the final version of the application fulfills all criterias initially presented to the group.

The final product is simple and easy to use, which means it would be easy for anyone to use the application if one should desire so. It would require some basic knowledge of balance parameter and maybe a read through T. Aasen's article "The Turning Point for Maintaining Balance" in order to understand this new parameter.

The application does hold a lot of potential and could prove very useful to a lot of groups. A simple example could be a teacher who wishes to gain insight into students' balance and use it as a help to catch potential challenges early.

REFERENCES

Aasen, T, Nordahl, SHG, Goplen, FK, & Knapstad, MK. (2019). The turning point for maintaining balance. J diagn tech biomed anal 8: 1. Volume 8• issue 1• 1000138• page 2 of 3•. *Turning point (seconds) of center of pressure (COP) Healthy Patients T-test EO 0.49 0.77 0.018* EC 0.43 0.7 0.012* F-EO 0.42 0.55 0.052 F-EC 0.43 0.45 0.589 Table, 1*, 3.

Bartlett, Harrison L, Ting, Lena H, & Bingham, Jeffrey T. (2014). Accuracy of force and center of pressure measures of the wii balance board. *Gait & posture, 39*(1), 224-228.

Leach, JM. Mancini, M. Peterka, RJ. Hayes TL. Horak, FB. (2014) Validating and Calibrating the Nintendo Wii Balance Board to Derive Reliable Center of Pressure Measures. *Sensors(Basel),* 14(10), doi: 10.3390/s141018244

Lou, C., Pang, C., Jing, C., Wang, S., He, X., Liu, X., Huang, L., Lin, F., Liu, X., & Wang, H. (2018). Dynamic Balance Measurement and Quantitative Assessment Using Wearable Plantar-Pressure Insoles in a Pose-Sensed Virtual Environment. *Sensors (Basel, Switzerland)*, *18*(12), 4193. https://doi.org/10.3390/s18124193

Hansson, E. E., Beckman, A., & Håkansson, A. (2010). Effect of vision, proprioception, and the position of the vestibular organ on postural sway. Acta oto-laryngologica, 130(12), 1358-1363.

Weaver, TB. Ma, C. Laing, AC (2017) Use of the Nintendo Wii Balance Board for Studying Standing Static Balance Control: Technical Considerations, Force-Plate Congruency, and the Effect of Battery Life, *Journal of Applied Biomechanics,* 2017(33)

LINK TO WEB SOURCES

GitHub user bendermh. (2019). RombergLab. Available from https://github.com/bendermh/RombergLab [Accessed 24.01.20]

Maven. (2020). What is maven. Available from https://maven.apache.org/what-is-maven.html [Accessed 30.03.20]

Scrum.org. (2020). What is scrum? Available from https://www.scrum.org/resources/what-is-scrum [Accessed 14.05.20]

Tutorialspoint. (2020). Design patterns - MVC pattern. Available from https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm [Accessed 24.04.20]

Tutorialspoint. (2020) JavaFX. Available from https://www.tutorialspoint.com/javafx/index.htm [Accessed 21.03.20]

Youtube user Genuine Coder. (2016). Javafx UI responsiveness. https://www.youtube.com/watch?v=fl6Wp1I9wHQ [Accessed 15.05.20]

APPENDIX

Appendix 1 - Risk analysis

Risk	Possible undesired scenario	L	C	RF	Measures
Application is not done	The application is not useful to the project owner if parts of the application is unfinished.	1	4	4	Work continuously with Scrum and make sure all main components of the application are done.
Misunderstanding	The team has misunderstood the project, the task at hand or which requirements needed for finished solution. There are many cases where misunderstanding could cause trouble, whether it be inside the group, with the project owner or the uni.	2	3	6	Proper communication from everyone working on the project is needed. The team took initiative and created a messenger group with project owner, supporters, supervisor and the group itself to minimize misunderstandings or eliminate them completely.
Not able to write report in time	The team focuses too much on development and postpone writing.	1	5	5	The team agreed to make sure one day a week is reserved for writing only.
Corona measures	Due to Corona measures, meeting up in person to discuss the project and ask questions is difficult, if not impossible.	5	2	10	Vise fremdrift og diskutere videre om prosjektet må gjøres online. Eventuelt spørre om hjelp må også bli gjort online gjennom f.eks skype, teams eller messenger. Presenting progress, discussing details and asking questions has to be done online. As mentioned the group already made a messenger group which will serve as this purpose. Meetings will be conducted through the online service teams.

Merge conflicts	Unnecessary time used to resolve merge conflicts.	2	2	4	Use git branches to minimize merge conflicts.
Expectations are too high	There is a possibility that the finished product does not match the project owner's expectations.	1	4	4	Communication with the project owner is important, keeping each other updated and discussing the project often.
Hardware is not good enough	Wii Balance board is broken, or not good enough to use.	1	5	5	Project owner will supply a new Wii Balance board.

The risk analysis is mapped out with potential risks and scenarios that could have an impact on the project as a whole. This analysis measures risk using a scale from 1-5 where both the likelihood of said risk to occur and the consequence of said risk is given a number. The Risk Factor is calculated using the formula:

$$L\ (Likelihood)\ *\ C\ (Consequence)\ =\ RF\ (Risk\ factor)$$

As a rule of thumb, risks with 10 and below should be handled easily with zero to little effort required. These risks will also not be much of a threat to the project as a whole considering the group solves these fast. If the Risk Factor is greater than 10, the group has to develop a plan to combat these risks. The group made this risk analysis to lay out plans for potential risks that could impact the project and to show the severity of potential risks that could happen. It is possible that the group will encounter some unpredictable risks and eventual solutions to these problems will come as the risks happen.

Appendix 2 - GANTT chart

Week:	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Task																				
Requirements					Exam	Exam					Easter									
Startup meetings					Exam	Exam					Easter									
Domain Introduction					Exam	Exam					Easter									
Decide technology					Exam	Exam					Easter									
Write Blog					Exam	Exam					Easter									
Write Report					Exam	Exam					Easter									
Pre-Project Report					Exam	Exam					Easter									
Design MVP					Exam	Exam					Easter									
Implement MVP					Exam	Exam					Easter									
Test MVP					Exam	Exam					Easter									
Design Additional Features					Exam	Exam					Easter									
Implement Additional Features					Exam	Exam					Easter									
Test Final Version					Exam	Exam					Easter									
Release Final Version					Exam	Exam					Easter									
Pre Project Presentation					Exam	Exam					Easter									
Status Report					Exam	Exam					Easter									
Report Draft 1					Exam	Exam					Easter									
Expo Poster					Exam	Exam					Easter									
Report Draft 2					Exam	Exam					Easter									
Hand in report					Exam	Exam					Easter									
Final Presentation					Exam	Exam					Easter									
Expo **(Cancelled)**					Exam	Exam					Easter									

Appendix 3 - User manual

DISCLAIMER

The Software and code samples available on this website are provided "as is" without warranty of any kind, either express or implied. Use at your own risk.

The use of the software and scripts downloaded on this site is done at your own discretion and risk and with agreement that you will be solely responsible for any damage to your computer system or loss of data that results from such activities. You are solely responsible for adequate protection and backup of the data and equipment used in connection with any of the software, and we will not be liable for any damages that you may suffer in connection with using, modifying or distributing any of this software. No advice or information, whether oral or written, obtained by you from us or from this website shall create any warranty for the software.

We make makes no warranty that

- the software will meet your requirements
- the software will be uninterrupted, timely, secure or error-free
- the results that may be obtained from the use of the software will be effective, accurate or reliable
- the quality of the software will meet your expectations
- any errors in the software obtained from us will be corrected.

The software, code sample and their documentation made available on this website:

- could include technical or other mistakes, inaccuracies or typographical errors. We may make changes to the software or documentation made available on its web site at any time without prior-notice.
- may be out of date, and we make no commitment to update such materials.

We assume no responsibility for errors or omissions in the software or documentation available from its web site.

In no event shall we be liable to you or any third parties for any special, punitive, incidental, indirect or consequential damages of any kind, or any damages whatsoever, including, without limitation, those resulting from loss of use, data or profits, and on any theory of liability, arising out of or in connection with the use of this software.

Connect Wii Balance Board

The Wii Balance Board is necessary for the application to work. In order to connect the board follow the following instructions:

1. Push the “Connect” button top left

2. Press the red button on the Wii Balance Board
3. The status bar should read “Connected”

Once connected there should be a live plot on the COP plot showing the current COP reading.

Otherwise if the display reads any of the following:

- Failed, try again ->
    - You may have wrong bluetooth settings or drivers. Check www.bluecove.org for compatible software.
    - Pairing may have failed once -> try again.
    - Program may have stopped working -> quit program and restart
- Disconnected ->
    - Board was connected but lost connection
        - Try to connect again
        - Try to change batteries in the board
        - Try to restart the program

NOTE: There is no need to disconnect the board when done.

NOTE: The connection process is necessary on every connect. The application keeps no track of previously connected boards.

Record a test

In order to use the application and record a test follow these steps:

1. Select a test duration

    This is the amount of seconds a test will record data.

2. Select number of tests

    This decides how many tests are run after one another.

    NOTE: each test will have the same duration defined in step 1.

3. Click “Start test”.

Once the test has started you should see the X/Y split screen and live recordings being plotted on the plots.

On the completion of a test the *Result area* should update and display the parameters from the test. There should also be a new button in the area *Switch test.* See following chapter how this is used.

If more tests are to be run a countdown will be displayed with red numbers. This is the countdown to the start of the next test.

Switch Test

If more than one test was recorded there should be multiple buttons in the *Switch test* area. The buttons are labelled "Test X", where X is instead a number. The tests are ordered in increasing order, meaning the first test run will have the label "Test 1". By clicking this the result area and all the plots will be that of test number 1.

Switch views

The *Switch View* menu changes the plots being displayed in the applications main area.

COP: A COP (Centre-of-Pressure) plot showing the test plot and the current COP reading.

X/Y: COP plot but split into X and Y axis for better overview over the movement on the axis.

TP: Shows 4 plots related to the turning point:

- Top Left: Timeseries (Xi), is the time series generated from the COP plot
- Bottom Left: MSD (Mean Squared Displacement) of the timeseries
- Top Right: Slope, a double logarithmic plot of the MSD.
- Bottom Right: TP (Turning Point), the turning point plot.

Export Data

There are 2 ways to export data, through the two buttons in the result area labelled "Save Current" and "Save All" and through the "File" menu option located top left.

You can save (export) the data currently being displayed (Save Current - button) or save the data from ALL tests run in that batch into the same file (Save All - button).

NOTE: If you are in doubt to which tests will be saved, it is the exact same tests that are located in the *Switch test* area.

How to export data:

1. Provide a person ID. This can be anything.
2. Provide a test type. This can be anything as well, but should include information of how the test was condoned.

   NOTE: A predefined test type, Romberg, will automatically set the duration to 20 seconds and the amount of tests to 4. This can be changed manually after Romberg has been selected.

3. Export All test (Save All) or export the currently displayed test (Save Current)
4. A file-selector window should appear
5. Find location and click "Open"

   NOTE: On other operating systems the button to click may differ from "Open" on Linux.

6. The data should be located where you selected as a CSV file.

NOTE: The file is preset to be labelled personId_testType_DDMMYY_TTTT_ CEST.csv with the predefined person ID and test type.