Creating Good User Experience in a Hand-Gesture-Based Augmented Reality Game

Benny Lam, Jakob Nilsson

Examiner: Erik Berglund
External supervisor: Magnus Lundqvist
Upphovsrätt


Copyright

The publishers will keep this document online on the Internet – or its possible replacement – for a period of 25 years starting from the date of publication barring exceptional circumstances. The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility. According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement. For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page http://www.ep.liu.se/.

© Benny Lam, Jakob Nilsson
Abstract

The dissemination of new innovative technology requires feasibility and simplicity. The problem with marker-based augmented reality is similar to glove-based hand gesture recognition: they both require an additional component to function. This thesis investigates the possibility of combining markerless augmented reality together with appearance-based hand gesture recognition by implementing a game with good user experience.

The methods employed in this research consisted of a game implementation and a pre-study meant for measuring interactive accuracy and precision, and for deciding upon which gestures should be utilized in the game. A test environment was realized in Unity using ARKit and Manomotion SDK. Similarly, the implementation of the game used the same development tools. When it comes to the virtual objects, Blender was used for making the 3D models.

The results from 15 testers showed that the pinching gesture was the most favorable one. The game was evaluated with a System Usability Scale (SUS) and received a score of 70.77 among 12 game testers, which indicates that the augmented reality game, which interaction method is solely based on bare-hands, can be quite enjoyable.
Acknowledgments

Dear readers of this thesis. If it were not for Etteplan, Manomotion, and the guidance of Erik Berglund, our examiner, this thesis would not have seen the light of dawn. First off, we would like to thank Erik for his patience and for the email-conversations we shared. We would also like to thank our supervisor from Etteplan, Magnus Lindqvist, for constantly supporting us and putting his trust in us while letting us playfully explore augmented reality. We also want to thank Erica Dahlberg for introducing us to Etteplan in the first place, welcoming us with open arms to a place where we would spend over half a year working on our thesis. Not last nor least, Abraham Georgiadis from the Manomotion family, thank you for a new friendship, and for bringing ideas and being so helpful with getting us started with using the Manomotion SDK — you have saved us countless of hours. It would be wrong to not express our gratitude towards everyone who participated in our tests, we thank you all for contributing to this study. Finally, to our beloved family, friends, and non-friends. Without you guys, who knows what would have become of the two of us: Benny Lam and Jakob Nilsson. We might have been pro bowlers or intermediate cricketers, who knows...

Thank each and everyone of you,
/Whiteboy & Asian
Contents

Abstract iii
Acknowledgments iv
Contents v
List of Figures vii
List of Tables ix

1 Introduction 1
  1.1 Motivation 2
  1.2 Aim 3
  1.3 Research Questions 3
  1.4 Delimitations 3
  1.5 Thesis Outline 3

2 Theory 4
  2.1 Simultaneous Localization and Mapping (SLAM) 4
  2.2 Visual Odometry (VO) 5
  2.3 AR Frameworks 7
  2.4 Bare-Hand 3D Gestural Interaction 8
  2.5 Game Design 10
  2.6 Usability 14

3 Method 19
  3.1 Pre-study 19
  3.2 Game Implementation 25
  3.3 Evaluation 32

4 Results 34
  4.1 Pre-study 34
  4.2 Game Implementation and Evaluation 38

5 Discussion 41
  5.1 Results 41
  5.2 Method 43
  5.3 The work in a wider context 44

6 Conclusion 46
  6.1 Future Work 47

Bibliography 48
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Illustration of a 3DOF configuration in Apple’s ARKit SDK</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Illustration of a 6DOF configuration in Apple’s ARKit SDK</td>
<td>7</td>
</tr>
<tr>
<td>2.3</td>
<td>Example of a Flowboard</td>
<td>13</td>
</tr>
<tr>
<td>2.4</td>
<td>Usability problems found over number of test users</td>
<td>16</td>
</tr>
<tr>
<td>3.1</td>
<td>Unity add-ons that are supported by Manomotion</td>
<td>20</td>
</tr>
<tr>
<td>3.2</td>
<td>Supported hand representations of the Grab, Point, and Pinch gestures in a Man-oClass</td>
<td>21</td>
</tr>
<tr>
<td>3.3</td>
<td>ManoGestures of available trigger and continuous gestures in the SDK</td>
<td>21</td>
</tr>
<tr>
<td>3.4</td>
<td>Position of the Interactive Point (IP) relative to the hand in different hand poses</td>
<td>22</td>
</tr>
<tr>
<td>3.5</td>
<td>Coordinate system of the phone’s position</td>
<td>23</td>
</tr>
<tr>
<td>3.6</td>
<td>Trigger test of the Tap gesture</td>
<td>24</td>
</tr>
<tr>
<td>3.7</td>
<td>Continuous movement test of the Grab gesture</td>
<td>25</td>
</tr>
<tr>
<td>3.8</td>
<td>From left: Baldy, Pinky, Nivens McTwisp, Snevin McTwisp, and a Thor Candy</td>
<td>27</td>
</tr>
<tr>
<td>3.9</td>
<td>The probability distribution for normal occasions and when the player has lightning powers</td>
<td>27</td>
</tr>
<tr>
<td>3.10</td>
<td>The template of the cabinet in Illustrator, a 3D model of the cabinet in Blender, and the final form of the cabinet in Unity — all seen from the front</td>
<td>29</td>
</tr>
<tr>
<td>3.11</td>
<td>The cabinet in before and after texture is applied, and the final look — shown from a front-side view</td>
<td>30</td>
</tr>
<tr>
<td>3.12</td>
<td>A sequence of the particle system and text of the score when a mole is being hit</td>
<td>30</td>
</tr>
<tr>
<td>3.13</td>
<td>A sequence of the lightning effects, and a radial bar representing the duration of the effects when obtaining a Thor Candy</td>
<td>31</td>
</tr>
<tr>
<td>3.14</td>
<td>A screenshot of the lightning effects, and a radial bar representing the duration of the effects when obtaining a Thor Candy</td>
<td>32</td>
</tr>
<tr>
<td>3.15</td>
<td>Help menu consisting of a minimal description of the game</td>
<td>33</td>
</tr>
<tr>
<td>4.1</td>
<td>Average task completion time and precision of every tester using trigger gestures</td>
<td>35</td>
</tr>
<tr>
<td>4.2</td>
<td>Average task completion time and precision of every tester combined using trigger gestures</td>
<td>35</td>
</tr>
<tr>
<td>4.3</td>
<td>Average accuracy of every tester using trigger gestures</td>
<td>36</td>
</tr>
<tr>
<td>4.4</td>
<td>Average accuracy of every tester combined using trigger gestures</td>
<td>36</td>
</tr>
<tr>
<td>4.5</td>
<td>Average task completion time and precision of every tester using continuous movement gestures</td>
<td>37</td>
</tr>
<tr>
<td>4.6</td>
<td>Average task completion time and precision of every tester combined using continuous movement gestures</td>
<td>37</td>
</tr>
<tr>
<td>4.7</td>
<td>Average accuracy of every tester using continuous movement gestures</td>
<td>38</td>
</tr>
<tr>
<td>4.8</td>
<td>Average accuracy and precision of every tester combined using continuous movement gestures</td>
<td>38</td>
</tr>
<tr>
<td>4.9</td>
<td>SUS-score of every tester</td>
<td>39</td>
</tr>
<tr>
<td>4.10</td>
<td>Average SUS-score and standard deviation</td>
<td>40</td>
</tr>
<tr>
<td>5.1</td>
<td>Statement with most variation</td>
<td>42</td>
</tr>
</tbody>
</table>
5.2 Statement with most unanimous answers ........................................ 43
5.3 Statement with most unanimous answers ........................................ 43
List of Tables

3.1 Representation of the variables: upTimeMin, upTimeMax, and spawnDelay . . . . 28
3.2 SUS-score intervals and what grade they correspond to . . . . . . . . . . . . . . 33
4.1 Number of grades obtained . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39
Technology has advanced in a rapid manner making it difficult to get a perspective on how much things have changed. About seventy years ago, the first computer, ENIAC, was designed and used by the U.S. Army. ENIAC weighed more than 30 tons and occupied 167 square meters. In contrast to the past we have something remarkably smaller, weighing less, and can be held in a single hand — something of which we recognize today as a smartphone. Smartphones have changed the way people interact with their mobile devices. It is not just a tool for mere communication, but also for entertainment, documentation, navigation, and much more. Smartphones have become a tool of necessity in today's society.

As technology advances, the components in smartphones get better and better each year up until the point where we have smartphones with processors fast enough and cameras good enough for supporting a technology that lets us perceive completely computational objects in the world around us — a technology known as Augmented Reality (AR) [59]. Imagine what we can do with a technology that takes the world around us, places computer-generated content on top such that it resembles the real world, and lets the content interact with our world. There would be many fields of which such technology could be useful, e.g., in the maintenance and repair field a mechanic could see instructions of what to do next when repairing something outside of their knowledge area, in the medical field a student can practice surgery in a controlled environment, in the military field soldiers can receive positions of enemy locations received by an unmanned reconnaissance aircraft, and interior designers can see in advance if a piece of furniture would fit in the desired corner, or you can battle a giant alien on the way home from work.

To our advantage we have access to AR technology in our everyday lives now that smartphones have become so powerful. We create well-designed virtual content and use a touchscreen as the primary input for interacting with our smartphone. Howbeit, when it comes to interacting in a three dimensional environment we have to come up with a more natural and intuitive way before augmented reality will find its way into our daily lives and be adopted by the public. Perhaps we can interact with virtual objects the same way we interact with physical objects? Taking augmented reality in to the mobile world has resulted in people thinking of other ways to interact with the phone, such as using hand gestures as input for interacting in a three-dimensional virtual space. Imagine the possibilities when...
combining hand gesture support with augmented reality; it will change the whole game for how we implement, design, and use today’s applications.

1.1 Motivation

The project is carried out at Etteplan within the field of Augmented Reality, which is one of many different fields that the company is currently providing services within. An augmented reality system can be of two types, marker-based or markerless-based [29]. This study challenges state-of-the-art technologies by implementing a hand-gesture-based AR game based on the latter.

1.1.1 Etteplan

Etteplan is helping leading companies in the world in the manufacturing industry by providing design engineering services, technical documentation solutions and embedded systems and Internet of Things (IoT) services. Among the technical documentations, Etteplan is providing Augmented Reality and Virtual Reality (VR) services which can be used by their customers for training, troubleshooting, installation, or maintenance purposes. AR and VR can, for instance, display labels on a system to give a maintenance mechanic clear instructions of how to maintain a system, or supporting service engineers with repairing machines by providing the ability to look through big machines as if it was with x-ray. The VR and AR services are in a wide range when it comes to size and extent, with services that are used by big manufacturing companies as well as small applications that are used by Etteplan themselves on fairs and conventions to show the possibilities and potential of AR and VR.

1.1.2 Background

One of the applications that Etteplan currently has is mainly used on fairs to show customers possibilities with augmented reality. As of now, their application is marker-based which means that they have to carry a physical object to an exhibition in order to demonstrate their application. Using an AR application on a mobile device could be cumbersome if placing a 3D virtual object is dependent on image tracking since it would require the user to carry the marker used for calibrating the camera pose in addition to a tablet or a phone. Of course marker-based AR could come in hand sometimes, depending on the purpose of the application. Etteplan’s current application is developed in Unity with the Vuforia SDK. The benefits of using Vuforia is that the application can be built on both Android and iOS while the drawback is that Vuforia does not support simultaneous localization and mapping (SLAM). Etteplan has an attitude that brings fresh perspective to engineering challenges wanting to adapt with more modern techniques. This project has emerged as a result to creating a SLAM-based AR application.

Google and Apple have both recently launched their own libraries for SLAM-based AR which do not require image tracking for placing an object in the room. The disadvantage with ARCore (Google) and ARKit (Apple), though, is that they are platform specific. However, both ARKit and ARCore can be integrated with Unity which is a huge benefit. Being one of the biggest game engineering platforms, one of the advantages of using Unity for game development is the good developer community. Another advantage is that many software development kits (SDK) are well integrated with Unity; thus, making Unity a versatile tool for developing games as well as exploring possibilities with AR.
1.3 Manomotion

A few months after Apple announced and released the beta of ARKit, a computer vision company called Manomotion released their SDK for supporting hand gesture input with ARKit. Manomotion provides a real-time 3D gestural analysis framework which allows realistic gesture control for augmented and mixed reality (MR) applications. The company is based in Stockholm, Sweden and Palo Alto, US. The core technology that lays the foundation for the framework has been refined through seven years of research in gesture technology, now making it possible to achieve a precise hand tracking and gesture recognition in 3D-space simply by using a 2D-camera that is available on any smart devices today. Their SDK enables us to develop an application that can be controlled with the gesture of a hand.

1.2 Aim

The aim of this project is to develop a hand-gesture-based augmented reality application that Etteplan can use on exhibitions to give customers an insight on the potentials of AR and hand gestures in a mobile application. The purpose of the application is, in addition to showing that Etteplan is being in front and is fast to adapt to cutting edge technology, to demonstrate the latest technology in the field of augmented reality and using hand gestures as input for interacting with an application. The thesis also consists of a pre-study which intention is to help examining and selecting potential hand gestures for usage in the final application.

1.3 Research Questions

Based on the introduction and background, the research question that will be answered in this thesis is the following:

*How to design a hand-gesture-based augmented reality game with good interactive accuracy?*

1.4 Delimitations

Apple’s ARKit has been the chosen framework for developing an augmented reality application that is hand-gesture-based. The Manomotion SDK will be used for supporting hand gesture control in ARKit as it is the only one available today. The thesis is limited to the gestures provided in the freemium plan, meaning that no gesture definitions of our own will be explored as the enterprise plan is based on a yearly license.

1.5 Thesis Outline

Chapter 2 describes the related background theory, followed by Chapter 3 that outlines the working process. In Chapter 4 the results of the thesis are presented. Chapter 5 discusses the results and the chosen methodology, followed by a presentation of the conclusion in Chapter 6.
The concept of enhancing the information we can perceive with our senses by blending virtual graphic models with what we see in the real world is believed to be coined by a former Boeing researcher, Tom Caudell, in 1990 [29][35]. As mentioned in the introduction there can be two types of Augmented Reality systems according to Johnson: marker-based and markerless-based [29]. Applications that are based on markers must have the camera pointing at a physical object that has been registered as a marker in order to render a virtual object on top of it [11]. In contrast to marker-based AR, markerless applications function anywhere as they are not dependant on the need for supplemental reference points [29]. However, for achieving a pleasant experience with markerless-based applications it is necessary to have good models of the AR system and the environment which is why markerless AR require more components than a marker-based one. A tracking system is needed for markerless applications since they make use of positional data, such as a GPS (Global Positioning System), a compass, or image recognition where input from a camera is compared against a library of images to find a good match [29]. The use of a tracking system helps with solving the problem of estimating properties of the environment and the navigation [53].

2.1 Simultaneous Localization and Mapping (SLAM)

A topic that emerged in the field of mobile robotics has become increasingly important within the computer vision community, especially in the AR and VR area, is about Simultaneous Localization and Mapping (SLAM). The chicken-and-egg problem occurs in mobile robotics as well as in AR. This is because a map is needed for localization and a pose estimate is needed for mapping. It is contradicting trying to construct or update the map of an unknown environment while simultaneously navigating through it using the map. Marker-based tracking, for instance, is not SLAM as the representation of the marker is known beforehand [17][27]. Note that when we are referring to a “SLAM system” we mean “a set of algorithms working to solve a SLAM problem”. There are many ways to implement such a system and different algorithms use different kind of hardware to approach the problem. Since every mobile phone today is equipped with a camera, using it to estimate the pose of the phone and mapping an unknown scene with only vision is a way of solving the problem. This is called visual SLAM, also known as V-SLAM.
2.2. Visual Odometry (VO)

In the robotics field, there may be a set of cameras used to construct a map and tracking the robot. However, a phone is often merely equipped with a single camera that either faces the back or the front of the phone. A work on a monocular SLAM system is presented by Davison in which he uses a single camera for achieving real-time SLAM [14]. Another one is presented by Eade and Drummond but uses a different algorithm [20]. Both of the works by Davison et al. (MonoSLAM [15]), and Eade and Drummond can be seen as adaptations of SLAM systems in the robotics domain which are EKF-SLAM [54] and FastSLAM [38], respectively. The work that Davison presents is accurate and robust but cannot scale to large environments since SLAM methods that utilize the extended Kalman Filter (EKF) is known for the order of the computational complexity of the filter update, which is $N^2$, where $N$ is the number of features in the map [14]. In contrast, Eade and Drummond’s work on a scalable monocular SLAM is based on FastSLAM rather than EKF-SLAM, meaning they use a particle filter instead of the EKF to estimate the pose of the camera, achieving a $O(M \log N)$ complexity where $M$ is the number of particles and $N$ is for landmarks [20].

Although both systems are robust, neither of them provide the robustness desired for AR use. As a result, Georg Klein and David Murray’s work on parallel tracking and mapping (PTAM) emerged [31]. They provide a solution which motivates a split in tracking and mapping, arguing that most computers now come with more than one processing core making it convenient to separate tracking and mapping into individual threads. This reduces the number of video frames needed to use for mapping, and the tracking thread is able to perform more thorough image processing which increases performance. PTAM is the first SLAM system designed for mobile AR and was developed in 2007. PTAM was first tested on an iPhone 3G in 2009 with some modifications, as one of the main challenges back then was the lack of processing power of the phone; it was a single core processor [32]. Consequently, as years go by, technology advances and we now have more powerful mobile devices than ever. The iPhone X comes with a six-core 2.39GHz processor compared to the 412MHz single processor of the iPhone 3G [24][32].

2.2 Visual Odometry (VO)

Visual odometry is another topic that has received a lot of attention when it comes to mobile robotics and computer vision. As mentioned before, SLAM is the process of a robot trying to localize itself in an unknown environment while simultaneously building a map of this environment without any prior information and with the aid of some type of sensors. Whereas VO is the process of estimating the egomotion of an agent by solely using the input of a single or multiple cameras attached to it. Now this may sound a lot like V-SLAM, however, the goal in V-SLAM is to obtain a global and consistent estimation of the robot path which includes keeping track of the environment map. The environment map is used to detect when the robot has returned to a previous visited area. This is called loop closure and is used to reduce the drift in the map and camera path. In contrast, in VO the goal is to recover the path incrementally, pose after pose, and potentially optimizing over the last n poses of the path (referred to as windowed bundle adjustment). While SLAM cares for the global map consistency, VO is only concerned for local consistency in the trajectory and the local map is used to achieve a more accurate estimate of the local trajectory. [52]

The first known VO real-time implementation on a robot was described by Hans Moravec in the 1980s. His approach to the problem of estimating the egomotion of a robot was by observing a sequence of images that was captured using a single camera sliding on a rail to give a stereo vision. A total of nine images were captured by the camera. First, the robot would stop and capture an image and extract the features from the first image, then the camera would slide on a rail in a perpendicular direction with respect to the motion of the
2.2. Visual Odometry (VO)

robot until nine images were taken. Features gathered from the pictures were matched and used to reconstruct the 3D structure which was then used to obtain the camera motion transformation by aligning the 3D points from the different locations \[66\][41]. Moravec’s work was later extended by Matthies and Shafer who instead of using a scalar model, approached the problem with error modelling using 3D Gaussian distributions \[37\]. Matthies et al., continued to further develop previous work with minor variations and modifications to help improve the robustness and accuracy for mobile robot navigation \[10\]. These papers laid the foundation for applying VO in the ongoing space mission on Mars – this is what VO is most famous for. The two rovers that were sent landed on the surface of Mars in January 2004 with the purpose of exploring the geology and surface of the planet. Cars and drones are other robotic applications where VO can be used for its ability to provide localization, in perhaps GPS denied situations \[47\][22].

2.2.1 Visual Inertial Odometry (VIO)

In retrospect, additional sensors may be used to aid in obtaining an even more accurate estimate of the agent’s egomotion. Inertial measurement units (IMUs) can be utilized within the VO system and is then referred to as visual inertial odometry (VIO). Gyroscopes and accelerometers are typical IMUs used for VIO systems. The first VIO system is believed to have been invented at InterSense (a company focused on inertial tracking and sensor fusion technology) in the early 2000s by Naimark and Foxlin \[23\]. They present a selftracker small enough to wear on a belt, is low cost, easy to install and self-calibrate, and has a latency low enough to achieve AR registration.

By using a VIO system the possibility of providing the user with an immersive AR experience is present. This is because VIO makes it possible to construct the trajectory of the camera in six degrees of freedom (6DOF) which is important in AR and VR. What 6DOF refers to is the freedom of movement in a rigid body in three-dimensional space, keeping track of both translation and rotation. There are three types of translations and rotations that a body can do. A body can translate on the x, y, and z-axis (by moving forward/backward, left/right, or up/down) and rotate along these axes. Conversely, a 3DOF tracking system either tracks the rotation or movement changes of the device. Figure 2.1 resembles the tracking of 3DOF system where the AR illusion is maintained when the device pivots but not when it moves. Respectively, Figure 2.2 shows that the AR illusion is maintained when tracking in 6DOF, regardless of the rotation or movement of the device.

![Figure 2.1: Illustration of a 3DOF configuration in Apple’s ARKit SDK](image6)
In the end, the choosing between VO and V-SLAM is conditional on the tradeoff between performance and consistency, and simplicity in implementation. However, the best solution for tracking over a range of several hundred meters is VIO. If any system that is only based on vision matches its accuracy with VIO, it would still require more power (GPU or camera) than a VIO system and is what really matters when it comes to head-mounted displays (HMDs). So far, monocular VIO is the most accurate lowest power and lowest cost solution for immersive AR.

### 2.3 AR Frameworks

When it comes to what software development kit (SDK) to use for developing AR applications there are quite a few that are available for the public. The frameworks that we have chosen to acknowledge in our thesis are all very popular and compatible with Unity3D.

#### 2.3.1 ARKit

First up is the AR framework developed by Apple. This SDK is based on VIO meaning that an accurate 6DOF experience in AR can be achieved with low computing power. Apple announced and released the beta of ARKit for developers on June 5, 2017. ARKit was not available for regular consumers until September when it was released together with iOS 11. The key features or functionalities that ARKit provides are robust face tracking (using the TrueDepth camera which currently only exists on the iPhone X), accurate world tracking with VIO, horizontal plane detection, and light estimation which enables the correct amount of lighting to be applied on the virtual objects. All of these features together provide an immersive AR experience for the user. The lowest requirement for developing in ARKit is an Apple A9 processor. High performance processors enable fast scene understanding and allow content creators to produce detailed and compelling virtual content.

#### 2.3.2 ARCore

Right after ARKit was released, Google announced their approach towards AR which is the ARCore framework. As well as ARKit, ARCore was supported in Unity on the release day which was the on the 29th of August, 2017. The fundamental concepts of ARCore are motion tracking, environmental understanding, and light estimation. Motion tracking is what Google has decided to call their tracking system that enables the virtual objects to appear as
2.4. Bare-Hand 3D Gestural Interaction

if they were part of the real world. ARCore also detects feature points and planes for understanding the scene, just like ARKit. Both ARCore and ARKit provide similar features and give great AR experiences considering their ability to track virtual objects and merge them with the physical world. [25]

2.3.3 Wikitude

Wikitude has been in the market for more than 10 years. It was first launched in October 2008 and works on both iOS and Android. The Wikitude SDK has received many awards, among them are “best developer tool” and “best augmented reality browser”. The functionalities that the SDK provides are 3D recognition and tracking, image recognition and tracking, cloud recognition, location-based services, augmentations and visualization, and support for smartphones, tables, and smart glasses. The enterprise version of Wikitude costs more than 4490 euro per year. The free version can be used in development but comes with a big watermark covering the screen. [62]

2.3.4 Vuforia

Lastly, we have Vuforia which has been around dominating the market with marker-based AR applications for a long time. The SDK was released by Qualcomm in 2010 and acquired by PTC 2015. Vuforia seems to be the most widely used AR platform with over 425000 registered developers and has powered more than 50000 applications. The main features provided by Vuforia are image recognition and tracking, object recognition and tracking, marker tracking, scene tracking, cloud recognition, and text recognition. Vuforia is also a cross-platform framework that is well integrated with Unity. [60]

2.4 Bare-Hand 3D Gestural Interaction

Before the new augmented technology will be adopted by people, it is crucial to have an intuitive method to interact with it. The emerging popularity of AR makes it seemly to come up with a more natural way for interacting in a 3D environment. Since the existence of humans, our primary tool has been our hands. What if we could use our hands in the digital world the same way we use them in the real world when interacting with objects? Bare-hand 3D gestural interaction enables picking up an object in the virtual environment the same intuitive way it would be picked up in real life. Before focusing entirely on bare-hand interaction, it is important to acknowledge other methods that provide the ability to use hands for interaction with the virtual content to give some sort of perspective.

According to Mitra and Acharya, gesture recognition is the process of being able to perceive meaningful expressions of motion by a human which involves the hands, arms, head, face, and/or body [40]. To capture information regarding different hand poses some type of sensors are needed. Typically, these sensors include wearable sensors such as data gloves or external sensors such as video cameras. Suarez and Murphy [56] explain that while data gloves can provide accurate measurements of the hand pose and movement, they require a large amount of calibration, restrict natural hand movement, and are usually very expensive. On the other hand, video-based hand gesture recognition (HGR) addresses these issues but raises another problem regarding location and segmentation of the hands. The challenge in segmenting the hand from the background with a single RGB camera occurs particularly when there are illumination variations, occlusions, rapid motion of the hand or when other skin-colored objects interfere with a scene.

Instead of using the conventional RGB-cameras, color-depth cameras (referred to as RGB-D) may play a role in facilitating HGR in low-light or unpredictable environments where other
skin-colored objects are present (such as a face) \cite{56}. In computer vision, depth-cameras have been utilized for several years but have had a limited applicability due to its high price and poor quality. The release of low cost RGB-D sensors such as the Microsoft Kinect – that can provide high-quality depth images, addressing issues such as noisy background and lightning changes – has created a revolution in gesture recognition \cite{48}. Depth images can also be extracted from stereo video cameras instead of directly being sensed using a depth camera \cite{56}.

Finally, now that we have mentioned a few techniques for achieving HGR, all of them may not be suitable in a mobile environment. As mentioned before, using fiducial markers would require yet another physical component to be brought with when using the AR application. The motivation for not using data gloves is the same as diverging from marker-based AR – it is cumbersome. In addition, a glove-based solution hinders the naturalness and ease of the interaction between a user and a machine \cite{40}. To our knowledge, the iPhone X is the only phone (by the time writing this) that has a depth-camera and is located on the front of the phone which is not suitable for hand-gesture-based AR. Also, due to size and power limitations depth-cameras such as the Kinect are only available on stationary systems \cite{49}. Dual-cameras on the other hand, have become popular recently but is not widely adapted yet. Nevertheless, the majority of smartphones in the market today are equipped with a regular camera making HGR-technology based on RGB cameras accessible to everyone with a smartphone.

2.4.1 Manomotion SDK

Through 7 years of research, the technology developed by Manomotion introduces an interestingly new approach to gesture technology. The SDK provided by Manomotion allows accurate, nuanced, real-time hand tracking and gesture recognition with low processing power and requiring nothing but an ordinary camera that ought to exist on all smartphones \cite{36}.

The Manomotion SDK is based on the work that Yousefi proposes, which is a technology for a natural and intuitive bare-hand 3D gestural interaction in AR which solely uses the RGB camera as input, i.e. without the use of a depth sensor \cite{63}. Yousefi et al. presents a 3D gesture analysis using a large-scale gesture database that includes images of various articulated hand gestures \cite{65}. Bare-hand gesture recognition can be categorized into appearance-based and model-based approaches. Appearance-based approaches simplifies achieving real-time performance due to simpler 2D-image features but comes with the drawback that only simple hand gestures, such as detection and tracking of fingertips, can be handled. Conversely, 3D hand model based approaches, such as the Manomotion SDK, offer a richer description of hand gestures. However, background and lighting conditions need to be considered in order to acquire a good segmentation of the hand. The task is not trivial as a full analysis of the human hands — which are highly articulated, incorporating a total of 27 DOF on each hand \cite{21} — is required. This is when the gesture search engine plays a big role in making it possible to analyze the similarity between the database entries and pre-processed query to find a best match in real time with low computing power.

Matching the query images from the video input with the database containing all images is computationally expensive. Since the gesture search engine presented contains a large set of different hand gestures with possible variations of rotating, positioning, scaling, and deformations, the computational cost in matching queries with database entries would be really high if it were not for the index technology proposed by Zhou et al. \cite{68}. Manomotion has adapted the way Zhou et al. integrate powerful text retrieval tools together with computer vision techniques to improve the efficiency for image retrieval. Approaching HGR
as a large-scale retrieval problem instead of an appearance-based issue makes it possible to achieve complex hand gestures while simultaneously throttling the computational expense that otherwise would be high [64][65].

2.5 Game Design

In order to design a game, the designer must know what to design. So what is a game in terms of requirements and definitions?

2.5.1 Definition

Salen and Zimmerman define a game as a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome [50]. This definition is composed from a number of other definitions of different type of games such as: board games, card games, and computer games among many others. To achieve a clear and indisputable definition, the authors left out unnecessary parts and included the most important elements which are the following:

- **System**: Some sort of system has to be present. A system is a set of parts that interrelate to form a complex whole.
- **Players**: There has to be one or more participants interacting with the system.
- **Artificials**: The game should maintain a barrier towards time and space of the reality.
- **Conflict**: A central part of a game. If there is no conflict, there will not be a game. The conflict could be of different forms, such as a solo conflict with a game system or a social conflict between multiple players.
- **Rules**: The rules are limiting the system’s and players’ abilities which provides a structure to the game.
- **Quantifiable outcome**: The goal with the game should be quantifiable in some way. It could be a simple win or lose condition or a numerical score at the end of the game. This is often the element that differs a game from a less formal play activity.

The game part is however only one half of the game design, and the design part is certainly just as important for a game to function at all. Without the design, the game would just consist of random players and rules with no real meaning or purpose: The design is the game — without it you would have a CD full of data, but no experience [12].

To understand what a design is, Salen and Zimmerman define it as follows: Design is the process by which a designer creates a context to be encountered by a participant, from which meaning emerges [50]. The designer is in this case the individual game designer or the team of people that are developing the game. A game can also sometimes emerge from folk or fan culture in which the culture at large is considered to be the designer. The context of the game is the spaces, behaviours, narratives and objects of the game, and the participants are, as might be expected, the players that explore and manipulate the mentioned contexts by playing the game. Lastly, the meaning of the game is the outcome of the actions that the participants are making throughout the game. [50]

With both game and design defined, a definition for game design can be derived by combining the two, with the following result: Game design is the process by which a game designer creates a game, to be encountered by a player, from which meaningful play emerges [50].
2.5.2 Guidelines for Game Design

There is, however, no right answer to how a game should be designed and what steps there are to follow since it is such a complex process. There will be difficulties in the artistic process as well as in the technical process, and especially in the integration of the two. The process depends on a lot of variables and conditions, such as the personality and experience of the designer. Even though there are no step-by-step guide to follow strictly, there are a number of guidelines that should be considered when designing a game.

Goal and Topic

First of all, a well-defined goal and topic of the game should be present. The topic is the environment in which the player can reach the goal. This vital step is very often ignored by game designers even if it seems very obvious. The definition of the goal should express the effect that the game will have on the players. The importance of the goal will become more apparent later in the game design cycle. That is when the designer faces choices between certain features that can not be present simultaneously in the game. The choice should at this point simply be the one feature that brings the game the closest to the goal. [13]

As mentioned earlier, the topic is the environment where the goal is communicated through the concrete collection of conditions and events. For example, the goal of Tetris is to survive as long as possible using your stacking abilities in a non-space-consuming way, whereas the topic is different shaped blocks that gravitates towards the floor in an increasing speed. It is important to keep the topic subordinated to the goal. The other way around can keep the game designer from achieving the goal of the game and instead solely focusing on the topic. When selecting the topic, the potential topics’ ability to realize the goal of the game should be examined since there are a lot of pitfalls where the topic may interfere with the goal. Such a pitfall could for example be if the topic defines the blocks in Tetris as circle shaped. This makes it impossible to stack them in a non-space-consuming way, which interferes with the goal. [13]

Research and Preparation

With a selected goal and topic, the game has a purpose and an environment in which the game will play out. In order to make the game feel realistic and trustworthy, it is crucial that the designer knows most aspects of the topic and has done the proper research before starting the design [50]. These aspects could be the history of the topic, the motivation of any involved character, what excites current fans of the topic, and the relation between the characters, to name a few. While doing this research, the designer might find that the defined goal will not fit very well with the background of the topic and may have to adjust it to reach an acceptable level of authentic feeling [13].

Design Phase

The designing phase is, as the name reveals, where most of the design decisions are made. It is in this phase that the game takes some sort of shape and where the final product can be recognized. There are different parts of the design process, where some parts can not be revisited or changed once a decision is made. To distinguish these parts, Ernest Adams has divided the process into three major stages in his book Fundamentals of Game Design [3]:

- The Concept stage
- The Elaboration stage
- The Tuning stage
Each and every step consists of a number of design tasks. The *Concept stage* is the stage in where the irreversible designing decisions are made. These decisions will stick during the whole project. Changing them later would force a huge amount of the work that has been done to be thrown away. For example, when constructing a vehicle, one can change the color while the vehicle is still under construction, however, it would be too late to make the decision that a boat should be constructed instead of a car once the wheels are in place.

*Getting a concept* is the first design task. The concept is the designer’s general idea of how the game will entertain the player throughout the gameplay. The concept is affected by many different factors, one of them may be what genre the game should belong to. This part goes hand in hand with the *Goal and Topic* guideline that was explained earlier. A well-defined goal and topic facilitates the determination of what concept the game has.

The designer should also *Define an audience*. It is important to know what audience will play the game and find joy in it so that the designing decisions will always benefit that specific audience. When testing each design decision against the hypothetical representative player, any test should be performed by a person from the targeted audience to make sure the decision entertains the correct group of people.

*Determining the Player’s role* is the third design task. Most of the time when playing a game, the player is projected into a fictional world and playing a role in the game. There can be many various roles, such as an athlete, a detective or a driver. It is the designer’s job to define what the role of the player should be, which helps the player understand the objective of the game. A clearly defined role is therefore a vital part for the player to imagine the frame of the game and find the optimal path to achieve the goal of the game.

The last design task of the concept stage is to *Fulfill the player’s dream*. With the three previous tasks defined, the designer should take into consideration of how the game will fulfill the dreams of the player, whether it is dreams of creation, power or achievements. The designer has to make decisions about what challenges the player will face, what experiences the game will offer, and what actions the player will perform.

The second stage is the *Elaboration stage*. This is when all the general design decisions have been made and it is time to dig into the more specific designs. The elaboration stage is a process of iterative refinement, where the theoretical evolves into something concrete. The first task of this stage is *Defining the Primary Gameplay Mode*. The primary gameplay mode is the mode where the player will spend most of the time while in the game, typically 60-80 % or even more. A game like FIFA, for example, has one mode where the user acts as the coach, picking out the 11 players that are about to be on the field, but the primary gameplay mode is of course the soccer game itself, where the user acts as the players on the field. The primary mode is the most essential part to get right since this is the heart of the player’s experience and is therefore the gameplay mode that should be defined first.

The second task is *Designing the Protagonist* (given that the game has one). This is different from designing a static character since every action that the player is making will affect the character in certain ways. The player will spend a lot of time looking at the protagonist and it is therefore important to create a design that will make the player care about the character and identify with it. With the design of the protagonist, the designer can restrict the player’s freedom by choosing what actions to implement and what predefined actions the character should have.

*Defining the Game World* is the next task, but a lot of work is done if the aforementioned topic of the game is well-defined. The designer needs however to be more specific of where
the game is played out and how it should be presented at this point. This task comes with a number of different dimensions such as the environmental, temporal, physical and ethical dimension. Depending on if the game world is based on a real location the designer can use photographs and maps to illustrate a trustworthy and credible display of the world. If the game instead is played out in a fictional world, the designer has to have an own vision and imagination to create the game world. [3]

With a game world defined, the player needs some sort of challenge to achieve the goal of the game. To create those challenges, the designer needs to Design the core mechanics. The core mechanics is the fundamental play activity that the player is repeatedly performing during the gameplay [50]. There can be either one single core mechanic, like in Pong where the only core mechanic is to steer the paddle to bounce the ball towards the opponent, or multiple activities that are composed into a suite of actions, such as Pokémon where the player has to steer their avatar in the world as well as catch and fight pokémon in order to move forward in the game. It is the repeating of the core mechanics that creates patterns of behaviour, which manifest as experience for the player, and it is the core mechanics that makes the player make meaningful choices that will take the player forward in the game [50]. When players feel that a game is not fun to play, it is often the core mechanic that is liable.

As mentioned earlier, a game can include more than just one gameplay mode. If that is the case, the designer can now Create Additional Modes. This could be a wide range of modes, such as the aforementioned coach mode in FIFA, a start menu or a simple view of the high-score. The transition between game modes could be because of a player action or it could be an automatic transition that contributes to the natural flow of the game [2]. To not miss any potential modes, the designer can create a flowboard. A flowboard is a chart of different modes and the conditions that will cause the transitions between the modes. Figure 2.3 shows an example of a simple flowboard [2].

![Figure 2.3: Example of a Flowboard](image)

To keep a game from being boring, it has to have some sort of Level design. The level design gives the player an opportunity to get a sense of progress in the game instead of doing the same thing over and over with the same difficulty. There are different ways to implement this sense of progression, such as progression through completing tasks, progress as distance to target, progress as the character grows or progress as the player grows [4].

Depending on how large the game is, the designer might want to Write a story. The larger
2.6 Usability

the game is, the bigger would the need of a story become. The story of the game is often what interests and involves the player and what gives them the reason to continue playing the game since the player wants to know what happens next. There are a number of different ways in which the story can be integrated into the gameplay. It could serve as a reward when a transition between modes occur after completing a task or level or it could simply be told within the scene during gameplay. Either way, a well-written story will involve the player to a higher degree. [3]

The last stage is the Tuning stage. This stage is entered when the entire design of the game has been decided upon. This means that no more features can be added once this stage is entered. Instead, the designer should make small adjustments to the designing decisions made in the two previous states. This means polishing and tuning every possible feature until they are as good as possible. It is not always obvious when this stage should be entered and sometimes the designer is forced to enter it due to a set deadline. However, if time is not a factor, this stage should be entered when the designer feels like the design is complete and is happy with every design decision.

2.5.3 AR Game Design

When designing a game in AR, there are a few additional guidelines that the designer should think of, according to Wetzel et al. [61].

The first one is to Create meaningful content. When working with AR, the designer gains the ability to add a lot of visual features that the player will find exciting. However, there is no point in adding a feature just for the sake of it and get carried away with the infinite possibilities — every content should have a defined purpose. Following this guideline will also help the designer to Show reality. Adding too many AR features may obscure the real world, resulting in a design that feels clustered and messy. This will help the designer to Keep it simple and focus on clear and simple interaction schemes.

The guidelines mentioned above will also help to Use the real environment. This mostly concerns bigger AR games, such as street-based games, where the big distances between certain AR elements can make the player feel bored. This issue could be solved to a certain degree by making players take advantage of the real space. It could be as simple as encouraging the player to visit a real café as part of the game.

Lastly, the designer should Choose the tracking wisely. It is vital to realize and understand both the flaws and the strengths of the potential tracking methods to recognize the most viable one. This will facilitate the designing of the game so that the flaws of the method can be avoided and the strengths can be utilized.

2.6 Usability

To create a well-functioning and user-friendly application, the developer must know what the users of the application think about it. Is the application missing something? Could the interface make the application more self-explanatory? Is the interface too chaotic? There are many things that are not as obvious to the user as they are to the developer, which all affect the usability of an application.

The International Standard Organization defines usability as "the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" [1]. Steve Krug, on the other hand, is defining it as "making sure that something works well: that a person of average (or even
below average) ability and experience can use the thing — whether it’s a website, a fighter jet, or a revolving door — for its intended purpose without getting hopelessly frustrated" in his book Don’t make me think\[33\]. Usability is a quality attribute that evaluates how easy user interfaces are to use. Jakob Nielsen, co-founder of Nielsen Norman Group, defines usability with five components which embed some principles of UI design:

- Learnability - How easy is it for users to accomplish basic tasks the first time they encounter the design?
- Efficiency - Once users have learned the design, how quickly can they perform tasks?
- Memorability - When users return to the design after a period of not using it, how easily can they reestablish proficiency?
- Errors - How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- Satisfaction - How pleasant is it to use the design? \[43\]

There are other definitions of usability as well, but most of them consist of three common themes:

- There is a user involved
- The involved user is doing something
- The involved user is doing something with a system, product or other thing

Since this report will focus on creating an entertaining and user-friendly game, the usability of the game is of course essential. One basic and useful method of studying usability is through user testing, also known as usability testing\[28\], or usability evaluations. To test an application’s usability, a group of representative users are required for performing representative tasks with the design. The users are observed for when they succeed or have difficulties while executing their task. It is important that the users do the tasks themselves and not receive help through directing their attention to any particular part of the system, otherwise it can contaminate the test results\[42\]. Further, Stanley Dicks suggests that a software should be evaluated before a prototype is finished\[16\]. Dicks claims that a usability test that is performed on a complete application is not relevant and that a verification examination is better suited.

It is suggested that testing with five users is typically enough according to a research done by Jakob Nielsen and Thomas Landauer. They showed that the number of usability problems found in a usability test with n users is:

\[N(1 - (1 - L)^n)\]

where N is the total number of usability problems in the design and L is the proportion of usability problems discovered while testing a single user. Plotting the curve for L = 31% gives the result in Figure 2.4. The value of L chosen for the plot is an average across a big quantity of projects that they have studied. \[45\]

Figure 2.4 shows that testing with five users yields 85% of usability problems found while 100% is reached when testing with 15 users. The percentage of usability problems found is increasing rapidly when there are a few number of test users. However, the increment is lower and does not yield as much when the number of test users increases. Therefore, it is
2.6. Usability

Figure 2.4: Usability problems found over number of test users [44]

suggested to perform three studies with five users instead of a single study with 15 users. Doing 15 studies with a single user, unfortunately, does not generate the idea of diversity in user behaviour and insight into what is unique and what can be generalized. It will get harder to explore new problems and there may not be any new problems discovered. [44]

However, executing a usability test does not automatically lead to an improved application. If performed naively, by rule rather than by thought, the usability evaluation can be ineffective or even harmful. The choice of what evaluation method to use is therefore crucial and should emerge from a real problem or from a research question. A poor choice of evaluation method will induce an impending risk of trivial or meaningless results, encourage weak ideas, extinguish valuable concepts in an early stage of the developing process or give misleading feedback of how the end-user is using the design. In other words, performing user evaluations on new designs is a very useful and powerful tool as long as it is employed properly. Value is only provided when user evaluations are designed and carried out to help answer a meaningful research question. [26]

2.6.1 Usability in AR

Design guidelines for traditional user interfaces consist of many. However, these guidelines should not be entirely relied upon when evaluating usability in AR [19][55]. Most of the traditional guidelines were specifically developed for the WIMP (window, icon, menu, pointing device) interfaces. Howbeit, the methods and guidelines can sometimes be difficult to apply on an application that interacts with the 3D space around us as it sheds light to newer interaction techniques. In spite of the current available methods not being explicitly suited for AR environments, they can still contribute to discovering certain problems in AR applications [57]. To obtain a good overview of the problems a traditional usability method is facing when evaluating a virtual environment, Stanney et al. [55] present the following limitations:

- The multi-dimensional object selection and manipulation characteristics of 3D space is not represented by traditional point and click interactions.
- The traditional methods are not comprehensively addressing multimodal system output (visual, auditory, haptic).
- The traditional methods are not taking presence and after effects into account.
- Traditional performance measurements, such as time or accuracy, do not always fully characterize a virtual environment system interaction.
• The traditional methods do not consider the potential location and environment differences of the users testing the system.

With AR sharing many characteristics with virtual environments, these limitations could be applied to such interfaces as well.

As mentioned in the previous section, the choice of evaluation method will most likely define the value of the result and is therefore of highest priority. To simplify the decision of evaluation method, the developer should categorize and classify the different methods in order to find the superior one. In the paper *A survey of evaluation techniques used in augmented reality studies*, Dunser and Billinghurst [18] suggests that augmented reality user evaluations can be categorized into the four following types:

1. **Perception**: Experiments that observe simple tasks. The goal with these experiments is to understand how human cognition and perception operate in AR contexts.

2. **Performance**: Experiments that study user task performance within a specific domain of the AR technology. The goal with these experiments is to understand how the AR technology could affect underlying tasks.

3. **Collaboration**: Experiments that examine the general user’s communication and interaction between multiple collaborating users.

4. **System interface**: Experiments that have a more general approach. The goal with these experiments is to expose usability issues in the tested AR system or prototype.

In addition to categorizing user evaluation types, Dunser and Billinghurst claim that user evaluation methods in AR could be classified into five different groups in *Handbook of Augmented Reality* [17]:

1. **Objective measurements**: The goal with this type of method is to produce a repeatable and reliable assignment that will result in a quantitative observation. Common measures in such methods are accuracy (e.g., error rate), times (e.g., assignment completion times), test scores or position (of an object or the user).

2. **Subjective measurements**: These type of methods rely heavily on the subjective judgment of people. Ratings, rankings and questionnaires are the most used evaluation tools.

3. **Qualitative analysis**: Unlike the first category, this type does not put the results in numbers. Instead, the data is collected through interviews (structured, unstructured) or structured observations (direct or video observations).

4. **Non User-Based Usability evaluation techniques**: These type of methods do not require the end-users to perform the evaluation. Instead, cognitive walk-throughs or heuristic evaluations are used, as well as techniques that engage people that are not end-users (e.g., expert-based usability evaluations).

5. **Informal testing**: Considered as the least trustworthy type of method since it is only based on informal user observations or feedback that could have been gathered during a demonstration.

In order to achieve a high usability, the developer should identify the combination of category and class that will give the most trustworthy and solid answer to the given research question.
2.6.2 Usability in Hand Gestures

The usability in the field of hand gestures is usually defined by the five components mentioned in section 2.4, with an addition of one more component: the ergonomics [9]. Since the user is using the hand and arm to interact with the application, it is important for the needed gestures not to cause any physical stress or discomfort to the user. In order to create a favourable ergonomically designed hand gesture interface, Nielsen et al. list the following guidelines [46]:

- Avoid outer positions
- Relax muscles
- Relaxed neutral position is in the middle between outer positions
- Avoid repetition
- Avoid staying in the static position
- Avoid internal and external force on joints that may stop body fluids

As mentioned earlier, there are five more components that should be considered. To facilitate Learnability and Memorability, and to minimize the risk of Errors, the following characteristics should be aimed for:

- Simple to remember and execute
- Intuitive
- Icons and symbols should be logical towards the functionality [46]

The last two components, Efficiency and Satisfaction, are obtained by the sequencing and structure of the application. Sequencing involves the steps that are required to get from one state to the goal state, which for example could be a simple search action. The sequence would in this case be to click on the search field followed by filling in the desired search word and lastly pressing the search button. The sequencing becomes even more necessary in an application with gesture-based input since the number of possible commands to give is limited compared to more traditional input methods. [46]
3 Method

This chapter describes how our work was carried out to achieve the results. The study can be divided into three parts: pre-study, implementation, and evaluation. In the pre-study, an application for testing the accuracy and precision of various hand gestures was implemented. The pre-study can be described as an aid for determining which hand gestures the game would consist of. The implementation section narrates how the game is implemented. Finally, the evaluation section describes how the outcome of the gameplay-tests are assessed.

3.1 Pre-study

For the study, as mentioned in the theory chapter, the Manomotion SDK was used to achieve hand-gesture recognition. As the system would become an AR game developed in Unity and the only two AR frameworks that were supported by the Manomotion SDK were ARKit and Vuforia (see Figure 3.1), deciding which framework to use between the two was trivial as the only one that is based on monocular VIO is ARKit. Otherwise, if the Unity add-on for ARCore would have been available it would be harder to decide upon ARKit and ARCore. The ARKit plugin in Unity requires Unity v2017.1+, Apple XCode 9.3+ with the latest iOS SDK that contains ARKit framework, an Apple device that supports ARKit with iOS 11.3+ installed. The hardware used for the test was an iPhone X with iOS 11.3 installed. The platform used for developing was Unity 2017.4.1f1 and XCode 9.3.1 The frameworks used were Unity ARKit Plugin version 1.0.14, Manomotion Unity Plugin 1.9.1.5, and Manomotion Unity add-on version 1.3.

Before implementing an intuitive and accurate control mechanism for the game a few tests had to be carried out. These tests would lay the foundation for evaluating which hand gestures are considered to be best suited for specific tasks. However, to conduct the experiments an application had to be realized for that too. This testing application together with the results it generated can be interpreted as our pre-study.
3.1. Pre-study

3.1.1 Manomotion SDK Overview

There are two different analysis that are provided by Manomotion’s SDK: a static analysis of hand poses in single image frames, which is referred to as a ManoClass, and a continuous analysis of hand movements in a video sequence which is referred to as a ManoGesture.

A ManoClass is defined as the categories of supported gestures. These include three comprehensive classes of hand gestures: Grab, Point, and Pinch. Each of the classes include a range of states which are categorized into specific static hand poses (see Figure 3.2). There are a total of 14 states. For instance, the Grab class contains all the states transitioning from a “full open hand” to a “full closed hand”. All of the states immediately after the “full open hand” are classified as “open hand” poses. Additionally, other information provided by the ManoClass are the rotation of the hand, identification of both the left and right hand, and if it is the back of the hand facing the camera or the palm that is facing the camera. The latter, however, is only supported for the ManoClass that represents a Grab gesture.

A ManoGesture can be grouped into two categories: a trigger gesture, which is detected by analyzing a sequence of video frames that form meaningful actions such as “Tap” or “Click”, and a continuous gesture, which is consistent gesture detected in a sequence of video frames such as a “Closed pinch” or “Open pinch”. Figure 3.3 lists the names of all the trigger and continuous gestures available. To decide upon which of the available ManoGestures are easiest to perform, most accurate, and most precise, two different test environments were implemented: one that would put the triggers to test and another one that would challenge detection of a fixed hand pose.

3.1.2 Trigger and Continuous Movement Test

As previously mentioned, Unity was the intended developing environment to use for the hand-gesture-based augmented reality game. The ARKit plugin in Unity was downloaded from the Unity Asset Store which came with a simple scene that demonstrated the basic functionality of ARKit — a scene contains the environments and menus of a game. The Manomotion ARKit plugin was downloaded from Manomotion’s website and came with a copy of the same scene that demonstrated ARKit, the only difference worth noting is that the
3.1. Pre-study

Figure 3.2: Supported hand representations of the Grab, Point, and Pinch gestures in a ManoClass [36]

Figure 3.3: ManoGestures of available trigger and continuous gestures in the SDK [36]

scene from Manomotion came with a Hand Tracker Manager which is utilized to access the SDK so that hand gestures can be detected. The tests were built upon a copy of this particular scene.

The trigger and continuous movement tests require different tasks to be done but the test cycle remains the same. Each test has two scenes: the first scene is where the testing occurs, and the second scene presents the time it took for each task to be performed, the total time of all tasks, and the accuracy of the framework’s ability to detect the specific hand gesture. A complete test procedure consisted of five tests in total; two trigger tests that would review the Tap and Click triggers, and three continuous movement tests that would analyze the
3.1. Pre-study

A continuous gesture of Grab, Pinch, and Hold (closed hand with palm facing the camera). A total of 15 testers analyzed the gestures. The research was divided into two sessions, the first session consisted of six testers and the second session was done with the rest. As Nielsen suggested, it is better to perform multiple studies instead of a single one with 15 testers as the majority of the usability problems are already found when testing with five users [44].

For both the trigger and continuous movement tests, a physical hand would interact with virtual objects in an augmented world. When pushing a button or throwing a ball in the real world, the hand must first make contact with the object. In the same way, the logic in our tests require that you must make contact with the object that you want to interact with. Therefore an Interactive Point (IP) is introduced to detect the motor skill of our hands whenever we touch something and want to interact with that specific something. The shape of the IP is a cylinder with a size that is big enough to cover the palm of the hand. Its dimensions are 0.2x0.2x0.2 units which corresponds to 20x20x20 cm since the unit of measurement in Unity is meters. This cylinder follows the hand when a specific continuous gesture is detected and enables interaction with another virtual object when, for instance, the cylinder is colliding with that specific virtual object and a specific trigger gesture is detected. The rotation of the IP is set to be the same as the camera’s rotation. Figure 3.4 illustrates where the IP is located relative to the hand in different ManoClass families (i.e. point, pinch, and grab).

Haptic feedback is eliminated on the hand chosen to interact with the virtual objects. Of course one could make use of the phone’s vibration feature and whenever the hand touches the virtual object the phone vibrates. However, that feedback would be registered as unnatural and unintuitive as it is the wrong hand that feels the vibration. Instead, sight and sound are used for perceiving the virtual environment.

Trigger Test (Tap and Click)

The trigger test is based on four static spheres that are placed on a table (see Figure 3.6). One of the spheres is green. The task is to press the green sphere. Whenever the green sphere is pressed it changes color to red and then another sphere becomes green. The trigger test evaluates the Tap and the Click gestures which are both ManoGesture Triggers. Each tester will complete the test when the correct sphere has been pressed 20 times for each of the two gestures.

To make AR more immersive the spheres change color when they are touched (i.e. when the IP is colliding with them). If it is the correct sphere it changes to a slightly darker green and
3.1. Pre-study

Figure 3.5: Coordinate system of the phone’s position

if it is the incorrect sphere it changes to a different hue of red. Sound is played whenever the finger/fingers (depending on if it is the Tap or Click test) touches the sphere. Additionally, two distinct noises are played whenever a Tap/Click gesture is registered while touching the sphere, depending on if it is the red or green sphere.

For the Tap gesture test, the IP follows the hand when the hand is making a Point gesture (see Figure 3.2 and Figure 3.4) which the SDK refers to as a continuous Hold gesture. For the Click test, the IP follows the hand when it is making an Open Pinch gesture. Needless to say, the IP is invisible during the test sessions.

The test began with the researcher scanning the environment to find horizontal planes. When a plane was found the researcher placed the table on the plane and explained how to perform the gestures for the SDK to recognize the hand. The testee was then given the phone and could then interact with the spheres. Whenever they felt ready they pressed on the green sphere to start the test (this also starts the clock). During the test an observer counted what they interpreted as an attempt, from the testee, to perform the trigger action – the attempt are used for measuring accuracy. When the test was done the test application automatically switched scene showing the results for the respective trigger test. The results consisted of the time it took to perform each task and the total and average time of all tasks.

Continuous Movement Test (Grab, Pinch, and Hold)

The task for the continuous movement test is slightly different from the trigger test. This test requires the tester to grab a ball and put it inside a box, meaning that the virtual object is moved from point A to point B (see Figure 3.7). The IP follows the hand when it makes the gesture that the test requires, which can be the open hand (back), open pinch, or open hand (palm) (see Figure 3.2), referred to as Grab, Pinch, and Hold, respectively. Each tester should complete the task 10 times for each of the three gestures.

Similar to the trigger test, it is important for immersive AR that the perception and interaction are intensely tied. Sound is played when the hand touches the ball. It changes
color from red to a different hue of red while a sound is played to give the tester feedback. Sound is also played when the ball is grabbed (i.e. transition from open to closed hand (back/palm) or from an open pinch to closed pinch) or released (from closed hand/pinch to open hand/pinch). If the ball falls and makes contact with the box the user will hear a different sound as to when the ball falls on the table or floor. If the ball falls on the floor it is placed on the table again which is thought to achieve consistent test results since it would require more from a tester that keeps dropping the ball compared to a tester who does not. If the ball is placed in the box and remains there for 1.5 sec the task is completed and the position of the ball is set to where it was from the beginning of the test, simultaneously as a sound is played for resetting the ball. The time 1.5 sec prevents a task to be completed whenever the used drops the ball on the side of the box and it slides down to the table/floor.

The continuous movement test was initiated after the tester had completed the trigger test. Likewise, the tester was given the chance to obtain a feeling for grabbing, pinching, and holding the ball (depending on the test). When they were ready to start they would put the ball in the box which would also start the clock. Since the ball remains in the box for 1.5 sec for each task, that amount of time is also withdrawn from the results. An observer was present during this test as well, to count the number of attempts made for each gesture that was tested. After all ten tasks were completed the results were displayed and the procedure continued until all three gestures were studied.
Once the tests have been carried out, a compilation of the test results was made. As mentioned earlier in this chapter, the accuracy and the precision of the different hand gestures laid the foundation of the decision of which hand gesture to use in the final product. The accuracy was determined by dividing number of tasks completed with total tries, whereas the precision was determined by calculating the standard deviation of the task completion times. Additionally, the average time of completing one task (i.e., pressing one ball or putting the ball in the box once) was also taken into consideration.

### 3.2 Game Implementation

As mentioned in the theory chapter, the first step in implementing a game is to define a goal and a topic. The definition of the goal set its course with the help of a brainstorming session. The goal, that was agreed upon, was that it had to put stress on and penalize the player somehow, which in turn would require the player to be alert while performing a stressful task. With a determined goal, the brainstorming continued in order to find an already existing game that may have fulfilled the goal in which inspiration could be acquired. A game that fitted the goal description very well was the classic arcade game Whac-A-Mole.

Whac-A-Mole is an arcade game where mechanical moles randomly pop up from holes in a cabinet. The player’s mission is to force the moles back down into their holes by hitting them with a mallet. Each mole hit is rewarded with an addition to the player’s total score. If the player does not hit the mole fast enough, it sinks back down into its hole. The gameplay often starts with a slow pace but the speed is increasing gradually with more moles being above the hole at the same time and for a shorter period of time. The final score is in direct correlation with how many moles the player hits.

Since these types of games often can be found in amusement parks or other similar environments, the chosen topic of the game was to be able to present the player with an authentic feeling similar to being in an amusement arcade. As the game was going to be played in an augmented environment, the topic would be limited to the player’s physical space.

With the Whac-A-Mole-base, many of the design tasks in the Design Phase became self-explanatory. However, the audience needed to be defined. The targeted audience that
was agreed upon likes fast-paced immersive games and are keen to explore new technology. The core mechanics of the game are the same as Whac-A-Mole, i.e., to force the moles back into their holes. To make the game a bit different from the classic Whac-A-Mole, some new features and mechanics were implemented. In addition to the audience and the new mechanics, a level design also had to be implemented. Both processes are presented in section 3.2.1.

As it took quite some time, approximately two minutes, to build the scene on an iPhone X using a 2012 Macbook Pro, a separate scene was developed for running the game on the computer to debug and implement features for the game.

### 3.2.1 Game Logic

The core of the game is pretty straightforward, moles spawn randomly in different holes and are eliminated when they get hit or sink down the holes again after a random number of seconds, `Random.range(upTimeMin, upTimeMax)`. There are a total of 14 holes that the moles can spawn from. A mole can not spawn in a hole that is already occupied by another mole. A variable, `whichHole`, is set to a random integer between 0 and 13 to determine which hole to spawn the mole in. Before spawning a mole the variable is compared with a list, `activeMoles`, that keeps track of the holes that are currently occupied by another mole. If the hole is occupied, the variable is randomized again until a vacant hole is found. The hole is then added to the `activeMoles` list. A delay was also implemented to prevent moles from spawning simultaneously, however, that does not mean that moles can not spawn 0.1s apart from each other. The delay is determined by `Random.range(0, spawnDelay)`, where `spawnDelay` is two seconds.

To present the movement of the moles smoothly, linear interpolation between two vectors was used when spawning and when forcing the mole back down in the hole. There are five types of moles in the game and the type of mole to be spawned is determined before spawning it.

#### Moles and Power-Ups

A regular Whac-A-Mole game in the arcades only has one type of mole that gives score when being hit. However, in this game there are multiple moles. Some moles give score whereas others subtract or add seconds to the amount of time the player has left to play the game. There are also moles that give a boost to help the player obtain better score.

In Figure 3.8 the five moles available in the game are presented. The first two moles from the left, Baldy and Pinky, add 10 points to the score when being hit. The mole with a blue hat, Nivens McTwisp, gives a time bonus of five seconds but the mole to its right, Snevin McTwisp who has a bomb on top of his hat, subtracts five seconds from the timer when being hit. The last mole, a Thor Candy, takes the shape of a yellow sphere which is a power-up that forces all moles up from their hole, except for the mole carrying a bomb. The lightning powers last for five seconds and will destroy all moles that the player’s hand hovers over.

Each mole has different probabilities to be spawned, a probability that also depends on whether or not the player has lightning powers. In normal occasions, the probability that either Baldy or Pinky spawns is 70%, whereas Snevin McTwisp has a 20% chance to be spawned, and both Nivens McTwisp and a Thor Candy has a 5% to spawn. However, when lightning abilities are granted, Snevin McTwisp will not spawn at all. The allocated probability distribution is now instead 49%, 49%, 0%, 0%, and 2% for Baldy, Pinky, Nivens McTwisp, Snevin McTwisp, and Thor Candy, respectively. See Figure 3.9 for a visualization.
of the outcomes. The probability of the chosen outcome can be visualized as a dart being thrown on a piece of paper and seeing where it lands. Since there are five moles, the paper strip is divided into five sections which take up a fraction of the paper strip that corresponds to the probability that that specific mole spawns.

![Figure 3.8: From left: Baldy, Pinky, Nivens McTwisp, Snevin McTwisp, and a Thor Candy](image)

![Figure 3.9: The probability distribution for normal occasions and when the player has lightening powers](image)

**Level Difficulty and Score Management**

A game is only entertaining if there is a fine balance between challenges and skills. If the challenge is too high in relation to the player’s skill, he or she may feel frustrated or anxious while playing the game. The intention with playing a Whac-A-Mole game in AR should not contribute to boredom or anxiety. However, being too good at a game can eventually lead to boredom which is why a greater reward is granted in such cases.

In this game there are three variables responsible for challenging the player, upTimeMin, upTimeMax, and spawnDelay. As previously mentioned, the amount of time the mole stays up in the hole is determined by a random value between upTimeMin and upTimeMax, and a random value between zero and spawnDelay determines the delay of spawning a mole. The default scenario (level 1) sets these values to 2, 4 and 2, respectively. This corresponds to a mole being active between 2 and 4 seconds and waits up to 2 seconds before being spawned. The level changes if the player is skilled enough to eliminate 10 moles in a row, i.e., Blinky, Pinky, or Nevin McTwisp. There are 6 levels in total and to reach level 6 the player has to
3.2. Game Implementation

destroy 60 moles consecutively. Progressing to a new level makes the gameplay a little bit harder by increasing the pace of the game. The values that control the pace are presented in Table 3.1.

<table>
<thead>
<tr>
<th>level</th>
<th>upTimeMin (s)</th>
<th>upTimeMax (s)</th>
<th>spawnDelay (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1.75</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2.5</td>
<td>1.75</td>
</tr>
<tr>
<td>5</td>
<td>1.5</td>
<td>2.5</td>
<td>1.5</td>
</tr>
<tr>
<td>6</td>
<td>1.25</td>
<td>2.5</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3.1: Representation of the variables: upTimeMin, upTimeMax, and spawnDelay

There has to be a system that rewards a player for playing good as well. Whenever the player reaches a combo of 7 the score that each mole provides multiply up to a maximum of 5 (1x, 1.5x, 2x, 3x, 4x, and 5x), i.e., a single mole can give up to 50 points. Also, if the player obtains the Thor Candy, besides from it spawning profitable moles in all holes it also provides a quicker way to increase the multiplier.

3.2.2 Game Design

The desire of the theme and topic of the game was to make the player feel that he or she was situated in an environment that reminds of an amusement arcade. Although AR is not as immersive (or will ever be) as virtual reality (VR), one can always make an effort to implement a design that appears to surround the player in a somewhat realistic manner. Now, using ARKit to develop a game in AR already takes care of the part that makes it feel as if objects surround oneself the way they would in the real world. Combining hand-gesture-recognition with AR, the player can use their hand to interact with the Whac-A-Mole game similarly to an arcade in the real space. Among Unity, Blender 3D, Illustrator, and Photoshop were used in the process of making the design.

The Cabinet and Moles

The design of the cabinet set about in a traditional way — with a pen and a piece of paper. When a feeling of content was set, the design processed its way to the computer. Illustrator was used to create blueprints of the front and sides of the cabinet for speeding up the workflow when 3D-modelling in Blender. The blueprints were imported to Blender as images and the cabinet was modelled accordingly. Slight changes were made, such as the side compartment, designed for storing a coin that would be used to start the game, see Figure 3.10. Finally, when the model started to take shape of an arcade cabinet, texture was added to bring it alive. The texture of the cabinet was made up from a composition of posters and wallpapers that Etteplan would use to market themselves.

In amusement arcades, coins are required for playing a game. Likewise, a coin has to be put in the coin slot of the cabinet to start the AR game of Whac-A-Mole. The game is started and the timer is set to 60 seconds when the coin is placed in the coin slot. The coin slot was modelled in Blender as well but as a separate component from the cabinet. Moreover, holes which the moles would spawn from were modelled separately and would eventually, in conjunction with the coin slot, be put together with the cabinet in Unity. Simple models such as a coin, light bulbs, and a translucent window were made in Unity to complete the look, see Figure 3.11. The digital red font was used in an attempt for the cabinet to give a retro
3.2. Game Implementation

![Image of cabinet, mole, and UI elements]

Figure 3.10: The template of the cabinet in Illustrator, a 3D model of the cabinet in Blender, and the final form of the cabinet in Unity — all seen from the front.

The moles were designed in Blender as well (except for the Thor Candy as it is a simple model) but the process did not begin on a piece of paper this time. Instead, the mole was extruded and shaped from an icosphere until it resembled a mole. The body was reused to create other moles with different faces and accessories. The results can be seen in Figure 3.8.

**Miscellaneous Effects and Features**

Besides from a somewhat realistic cabinet and cute moles, other features were added to the game that would make the player feel more engaged while playing the game. The player would subconsciously recognize the details even though he or she may not think about it while playing. These details would for instance consist of particle systems, text that fades, or a radial bar that indicates how much time the player has left with lightning powers.

Needless to say, the implementation for the IP in the game was the same as in the trigger and continuous movement tests. Whenever the player hovers over a mole it becomes moderately lighter in color to show the player that the mole now can be hit if the required hand gesture is performed. When the player hits the mole a particle system that resembles an explosion is instantiated to demonstrate that the player has hit the mole successfully. To make it even more obvious, a text of the score that the mole was worth is also instantiated and slowly fades out while moving upwards. Figure 3.12 shows the effects in action through a sequence of pictures.
3.2. Game Implementation

Figure 3.11: The cabinet in before and after texture is applied, and the final look — shown from a front-side view.

Figure 3.12: A sequence of the particle system and text of the score when a mole is being hit.

Another case when a particle system was used was when the lightning effects were active, i.e., when the player got a Thor Candy. Feedback-driven experiences are always good to have in games and therefore, an orange radial bar was used to give the player a sense of how long the lightning effect’s duration lasted, see Figure 3.13. As mentioned in the Theory chapter, ARKit can estimate the light and apply correct amount of lighting on virtual objects which is taken advantage of when lightning powers are active. The light in the scene can be turned off or reduced when the player is in a dark room. However, when the lightning powers are active, the light in the scene is disabled automatically. Instead, the particle system has a luminescent material which gives away a flickering light when active. Other details are the lamps on the cabinet. They are luminescent and can distinguishably be seen when placing the cabinet in a dark room or when the lightning powers are active. Finally, the least
detail that the player may not notice at all is that the moles are always facing the player (i.e., the camera), which subconsciously makes them feel more alive.

Figure 3.13: A sequence of the lightning effects, and a radial bar representing the duration of the effects when obtaining a Thor Candy.

Sound

Background music and sound effects are used in games to increase the player’s sense of immersion. A total of 8 sound clips were used in the game. Some of them were custom-made whereas others were downloaded from a collaborative database of licensed sounds ([Freesound.com](http://Freesound.com)). The soundtrack of the theme was composed by Patrick De Arteaga and downloaded from his website patrickdearteaga.com. The song used is called Su Turno — a funny and happy royalty free track, perfect for an arcade with a Whac-A-Mole game. The music begins when the player inserts the coin to start the game. When there is only 10 seconds left of the timer, the player will hear a clock ticking, indicating that the game is about to end. Once the game finishes, the music also stops.

There are three different custom-made sounds for when the player scores, i.e. eliminates a Baldy or a Pinky. When the player is being dealt a penalty (Snevin McTwisp), an exploding sound from a bomb is made. While the lightning effect is active the player can hear a repeated electrical growl. The old game Street Fighter is used as inspiration for sound effects when the player gets extra time (Nevins McTwisp), the score multiplier increases, and for each 10th streak. For instance, when the player makes an uppercut in Street Fighter, he or she can hear that the narrator says in clear words “uppercut”. Similarly, in the Whac-A-Mole game the player will hear “bonus”, “multiplier”, and “combo” when appropriate.
3.3 Evaluation

**In-Game Tutorial**

Everyone who has been in an arcade is familiar with a Whac-A-Mole game, but since this is a modified version in an environment that is augmented and which can only be interacted with by the use of a real hand trying to touch something that in reality is not there, it might be frustrating playing the game. Since the fate of many mobile games today are often determined based on a first impression, not many people stick with it unless it is enjoyable. Therefore, an in-game tutorial was implemented to show the player how it is done. The tutorial begins once the player has found a horizontal plane and placed the cabinet in the real world. Since tutorials are boring in general and rarely done twice by the player, the player can always skip the tutorial if desired, see Figure 3.15. The key lesson to be learnt in the tutorial, besides from what each mole does, is that the correct gesture can be seen in the picture surrounded with a blue border.

If the player decides to skip the tutorial and during the gameplay gets confused with why the mole with the blue hat sometimes grants them time while occasionally also make them lose time, they can always press the button in the corner that resembles a question mark to get a clarification, see Figure 3.15. Pressing that button results in pausing the game and showing all the moles present in the game which would suffice to give the player a hint.

![Figure 3.14: A screenshot of the phone during the tutorial](image)

3.3 Evaluation

To figure out if the implementation of the final game appeals to the targeted audience and measure the usability of it, an evaluation was performed. The System Usability Scale (SUS) was used as a measuring metric since it is low-cost, can be used on small samples with reliable results and is valid. The testing of the game was carried out by 12 volunteers. They did not get any instructions other than that the game was a hand-gesture based AR game with the foundations from the classic arcade game Whac-A-Mole (and that their mission was simply to complete one round of the game). Afterwards, the testers were asked to fill out the SUS-survey.

The SUS gives a global view of subjective assessments of usability by using a 10 statement survey. Each statement in the SUS is answered by using the Likert scale, where every statement is rated with the level of agreement the tester is feeling. The Likert scale for a SUS
is rated from one to five, where one equals to strongly disagreeing and five equals to strongly agreeing.

Each tester receives a total SUS-score which is calculated in three steps. First, the value of each odd numbered statement is subtracted by one and then summarized to get the odd-score. Then, the value of each even numbered statement is subtracted from five and summarized to get the even-score. Finally, the total SUS-score is achieved by joining the odd and even-score together and then multiplying the sum by 2.5. This will result in a score in the range of zero to 100. The score is then graded as seen in the table below.

<table>
<thead>
<tr>
<th>SUS-score</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>81-100</td>
<td>A</td>
</tr>
<tr>
<td>73-80</td>
<td>B</td>
</tr>
<tr>
<td>60-72</td>
<td>C</td>
</tr>
<tr>
<td>51-59</td>
<td>D</td>
</tr>
<tr>
<td>&lt;51</td>
<td>F</td>
</tr>
</tbody>
</table>

Table 3.2: SUS-score intervals and what grade they correspond to

A score of 68 is often considered as the average score according to a huge study made by Jeff Sauro. An average score means that the tested system is decent but there is still room for improvements. A score above this average value is of course preferred, and studies show that a system with an A grade will make people recommend the system to their friends and family. A score below the average value of 68 might, however, indicate that improvements of the usability may be considered, and a system with an F grade should have a fix of the usability problems as a priority.
4 Results

This chapter presents the results achieved from the trigger and continuous movement tests. Additionally, results from the user tests on the finished hand-gesture-based AR game are also introduced.

4.1 Pre-study

Figure 4.1 illustrates the average task completion time of every participating tester as well as their precision when performing the trigger tests. The figure shows that the click gesture was performed faster than the tap gesture. Some testers even performed twice as fast with the click gesture than they did with the tap gesture. In addition to the lower times, the click gesture also had a superior precision rate (i.e., lower standard deviation) for 14 out of 15 testers. This becomes even more obvious when studying Figure 4.2 which illustrates the average task completion time and precision of the testers combined. The average task completion time for the click gesture was around 45% faster than for the tap gesture (3.12 seconds versus 5.67 seconds). It also shows that the precision was almost three times better for the click gesture.

In addition to the better task completion times and precision, Figure 4.3 shows that the click gesture excels in accuracy as well. Once again, every single testee performed better with the click gesture and achieved a significantly higher accuracy. The average accuracy for every testee combined was 41% for the tap gesture and 74% for the click gesture, which is shown in Figure 4.4. The graph also shows that the precision was about the same for the two different gestures.

During the trigger tests, all of the first six testers mentioned that they initially had problems with acquiring a good sense of the depth of the objects and that they, therefore, had some trouble getting the IP to collide with the virtual objects. In order to minimize the issue, the dimensions of the IP's cylinder were increased to 0.2x0.2x0.4 units so that the IP reached deeper along the z-axis, i.e., facing the camera (shown in Figure 3.5). Even though the results did not get any better in numbers after the calibration was done, there was no mentioning of the depth problem from the subsequent testers.
4.1. Pre-study

Figure 4.1: Average task completion time and precision of every tester using trigger gestures.

Figure 4.2: Average task completion time and precision of every tester combined using trigger gestures.

Figure 4.5 illustrates the average task completion time of every participating testee as well as the precision when performing the continuous movement tests. The graph shows that there is a large variation of the average task completion times between the testers which makes it hard to find a clear pattern of the measured times. This also applies to the precision which alters widely amongst the testers. The varying results between the testers did however result in a similar average task completion time when combining the results of the 15 testers, as shown in Figure 4.6. The testers completed the pinch and grab tasks with an equal average time of 13.34 seconds, whereas the hold gesture task was completed around 10 % faster with an average time of 11.89 seconds. The precision rates were also similar across the three different gestures, differentiating only by a maximum of 3 % from each other.
4.1. Pre-study

The accuracy of the continuous movement gestures was also very different across the testers’ results, as seen in Figure 4.7. The majority of the testers did, however, attain a slightly higher accuracy using the pinch gesture compared to the grab and hold gesture. Figure 4.8 shows that the accuracy of the pinch gesture was 72% compared to the accuracy the grab and hold gestures which was around 60%. Additionally, the figure shows that the precision was slightly better with the pinch gesture.

After investigating the results, it was clear that the click gesture was superior to the tap gesture in every aspect. In addition to the better numbers, the majority of the testers felt more comfortable using the click gesture and preferred it over the tap gesture when asked
4.1. Pre-study

Figure 4.5: Average task completion time and precision of every tester using continuous movement gestures.

Figure 4.6: Average task completion time and precision of every tester combined using continuous movement gestures.

about it after the test was completed. The click gesture was therefore the gesture of choice for action required tasks when implementing the final game.

The numbers in the result also showed a small advantage in favor of the pinch gesture after performing the continuous movement test. This was also the gesture that was preferred by most of the testers when asked which of the three that felt most enjoyable and convenient. Therefore, the pinch gesture was used when implementing any continuous movement tasks in the final game. This would also make the most sense since it would be counter-intuitive to use a continuous movement gestures different from the trigger gesture.
4.2 Game Implementation and Evaluation

The usability of the final game was evaluated using SUS as mentioned in section 3.3. The SUS-score of each and every tester is shown in Figure 4.9. The graph shows that most of the testers had a similar score with many of them having a score between 60 and 80. The highest SUS-score was however 92.5 while the lowest SUS-score was 50.

These different scores culminated into a result where all the five different grades — A, B, C, D, and F — were represented at least once. As seen in table 4.1 the most common grade was B that occurred 5 times, followed by grade C which occurred 4 times. The remaining
The grades A, D, and F only occurred once. The average score from all 12 testers resulted in a total SUS-score of 70.77, as seen in Figure 4.10.

Since the grades B and C were most common, the average SUS-score of the testers combined should be somewhere between 60 and 80. This is confirmed by studying Figure 4.10 which is showing that the average SUS-score of all testers combined was 70.77. This corresponds to a very strong C and is above the "global" average score of 68 presented by Jeff Sauro [51]. The final game had, therefore, a higher usability than more than 50% of the 500 systems Sauro tested in his study [51].

In addition to the average SUS-score, Figure 4.10 also illustrates the standard deviation of the test results, which in numbers was 11.5. This is a pretty low standard deviation in relation to the average SUS-score, which means that the precision of the score was high.
4.2. Game Implementation and Evaluation

Figure 4.10: Average SUS-score and standard deviation
5 Discussion

This chapter discusses the results we attained, our choice of method, and our work in a wider context. The discussion also brings up the practical use of the results and issues that need to be considered for hand-gesture based AR games to be trending among mobile gamers.

5.1 Results

This section mainly discusses the chosen hand gestures from the pre-study and why they were elected, as well as notable numbers and results from the game implementation and evaluation.

5.1.1 Pre-study

The results from the pre-study showed very convincing numbers when it comes to the trigger gestures. All 15 testers completed the task faster with the click gesture. As mentioned earlier, there were some issues with the sense of depth and where to put the hand or fingers to successfully touch the virtual object. The fact that the depth in AR can be an issue could probably explain the lower accuracy and higher completion time of the tap gesture, since the movement of the gesture was somewhat performed away from the camera.

The continuous movement gestures did not have a clear winner when looking at the numbers, which forced other non-measurable parameters to be taken into consideration. For example, the ball was sometimes misplaced on the testers forearm instead of the palm when performing the grab and hold gesture. It became clear that this behaviour occurred only when the tester wore shirts that did not cover their forearm, which made it hard for Manomotion SDK to obtain a clear hand segment.

It also made sense to choose the pinch gesture as the continuous movement gesture, since this was the counterpart to the chosen trigger gesture — the click gesture. This contributed to a more natural transition between the trigger and the continuous movement gestures during gameplay, which also contributed to a steeper learning curve compared to if the transition would be counter-intuitive.
5.1.2 Game Implementation and Evaluation

The most notable detail when studying the results of the SUS-survey (shown in appendix A.1) was that the statement that varied the most among the survey-answers was the "I thought there was too much inconsistency in this system", as shown in Figure 5.1. This was the statement that, ironically, was most inconsistent which can be interpreted from the dispersed result — each value of the Likert scale was chosen.

During our work with the implementation and by studying our testers, we noticed that the Manomotion SDK is somewhat sensitive to background noise. As mentioned in the Theory chapter (2.4.1) it is important to consider the background and lighting conditions for the SDK to work properly. Unfortunately, we were not able to conduct all of our tests at the same location which may have resulted in slight condition differences between some testers. We believe that this could have resulted in an environment with more noise for some testers which may have affected their feeling of inconsistency in the game.

The statements that had the most unanimous answers were "I thought the system was easy to use" and "I think that I would need the support of a technical person to be able to use this system". Figure A.1 shows that seven out of 12 testers agreed that they thought the system was easy to use. One of the five remaining testers strongly agreed to the statement whereas two could not decide to agree nor disagree, and the last two disagreed with the statement.

There were also seven testers that strongly disagreed that they thought they needed support from a technical person to use the system. Three testers disagreed with the statement and the remaining two testers did neither agree nor disagree with the statement.

As mentioned in section 4.2, the average SUS-score of the final game corresponded to a grade of C which is above the magic number of 68. This means that the usability of our system is solid and that it is working like intended. However, the C is also telling us that there is room for improvements to the system and that there are changes and/or additions to be made in order to obtain a higher usability.
5.2 Method

This section discusses the chosen method and whether or not the pre-study could have laid a better foundation for the study. It also discusses problems discovered when integrating the chosen techniques in the game implementation as well as if the chosen evaluation method was best suited for this type of study.

5.2.1 Pre-study

Most of the effort during the pre-study phase was spent on implementing the test application and performing the different tests. Since hand gesture recognition was a very vital part of the final game, it was important to obtain a reliable test basis. Research about integrating HGR technology with AR is relatively new to the world, robust SDKs for hand gestures likewise. Therefore, a good test environment was much needed in order the create a good picture of the different gestures and how the testers used the gestures. In other words, the lack of existing research and the importance of a reliable hand gesture recognition justified the large amount of time and effort put in the pre-study.
The results collected from the pre-study tests were for the most part quantitative data. The only qualitative data that was collected from the tests was a question regarding which gesture the tester thought was easiest and most convenient, which mostly was used as a complement to the quantitative data. To get an ever better overview of the different strengths and weaknesses of the hand gestures, more qualitative data could have been collected and used by asking the tester to answer more questions. This could potentially, however, have made it harder to find volunteers to perform the test since it would be more time-consuming to participate in the study.

5.2.2 Game implementation

As mentioned in 5.2.1, it was hard to find previous research about hand gesture SDKs combined with AR. Therefore, it was a challenge anticipating and avoiding problems that could occur when integrating the two technologies. During the implementation of the game, it was soon discovered that it was sometimes difficult for Manoomtion SDK to separate the background from the hand. While AR needs good texture to detect a plane in the real world, Manomotion requires a monochromatic colored background (different from the hand) to acquire an unambiguous hand segment.

Another problem that was discovered during the game implementation was that the early version of Manomotion used could be very sensitive to noise, such as real world objects in the camera view. When this problem occurred, it was a lot harder to interact with the AR objects, which of course did not add any satisfaction to playing the game in any way.

5.2.3 Evaluation

The methodology of using SUS as the evaluation for usability could have been altered to achieve other interesting results. As mentioned in 3.3, SUS is a low-cost method that gives relatively reliable results — the main reason it was chosen for this study. The results of SUS can however feel a bit vague due to the nature of the questions. It can reveal that the system is unnecessary complex, but often it does not reveal why the testers think it is complex and if there are only specific parts that are complex. To elude the issue and get a different type of results, a completely different evaluation method could have been used instead. Though, the best option would probably have been to use a complementary method in addition to the SUS. With an approach as such, the SUS could have pointed out problems with the system and the complementary method could then point out why those problems were caused and how to solve them to increase the usability. Unfortunately, time was a factor and only one method could be carried out in time.

5.3 The work in a wider context

As mentioned in the introduction, there is plentiful area for augmented reality to be utilized. Combined with gesture interaction technology it offers new and powerful ways to engage with the immediate surroundings. These state of the art technologies have opened up new chapters in the mobile gaming industry.

To challenge new technology, we have to use a vivid imagination to create something exceptional and fun to interact with. This is where ethical and societal aspects need to be discussed. The goal of a gaming company is, besides from growing, often to create a game that is hard to take a break from, i.e., creating an addictive game. Game addiction in a juvenile audience is pretty common these days and is hard to deal with, especially if a fun game can emerge in togetherness with a cool technology. A study by Zamani et al. showed that game
addiction made a negative impact on social skills of children. The higher their addiction to computer games was, the less were their social skills [67]. However, it would be unfair to say that the culprit is making a computer game more enjoyable than companionship with other human beings. Because, games can be used for educational purposes as well as contributing to a healthier lifestyle, e.g., by making exercise fun (especially with HGR and AR together).

An example of when an ethical aspect arises is when the ability to create customized avatars in AR games is made possible. For instance, an adult man can befriend a young boy in a game with intentions of preying on the kid by pretending to be the same age as him. It is a disturbing thought to battle as a parent when there is a risk that your child is interacting with someone online without the knowledge of the huge age gap between them.
Conclusion

The intention with this thesis was to make a remark and shine light on hand gesture recognition in coaction with augmented reality. How could HGR be used to facilitate interaction in an augmented world? This work was done to investigate how to create an AR game that was played using gesture recognition. The goal was to design a game that would maintain precise and stable interaction which can be reflected in our research question:

*How to design a hand-gesture-based augmented reality game with good interactive accuracy?*

This study found that the hand gesture recognition technology provided by Manomotion SDK works remarkably well together with ARKit. The developers at Manomotion have built a mature library that was easy to integrate in Unity which contributed to a helpful setup of a testing environment and game implementation.

The work was initiated with an application that was used in a test environment in which different hand gestures could be evaluated to find the one that was best suited for AR. The results from the tests revealed that users preferred the click gesture out of the two trigger gestures. For the continuous movement gestures it was the pinch gesture that took the price.

Succeeding to analyze the gestures for best practice, the work was carried on with implementing the actual AR game. It was decided that the classic arcade game Whac-A-Mole would lay the foundation of the game. However, it would consist of some alterations in mechanical behavior by using hand gestures as input. New features for the game was added as well, adapted for an AR environment.

After implementing the game, the final part of the study was to validate the game. This was done with the help of usability testing. A total of 12 testers were asked to validate it, simply by playing it and answering a survey regarding system usability. The evaluation conveyed satisfaction among the testers which would indicate that the game indeed was easy to interact with and fun to play. This would satisfy the research question that the game developed had good interactive accuracy, otherwise the evaluation would have foreshadowed it.
6.1 Future Work

Markerless AR and hand gesture recognition in mobile devices are both state-of-the-art technologies. This study has only scratched the surface of the large indefinite quantity of ideas to be realized with these cutting edge technologies. While it is easy and fun to evaluate new technology by creating games, it would be interesting to see other areas of usage as well.

Given more resources to continue this study, a multiplayer AR version of the game would probably have emerged, meaning that players would share the same space in the augmented world via internet. But, how one goes about taking up the challenge of merging multiplayer AR with HGR is for future work.

Another idea for future work is making the game accessible to multiple platforms, i.e. both Android and iOS devices, by utilizing AR Foundation which is a cross-platform API developed by Unity [58]. There is a noticeable demand of cross-platform games, especially for mobile devices since there really is not much that differs when playing games on different phones in contrast to playing the same game on PC and consoles. Cross-platform multiplayer games would bring people together and is that not what games are all about?

By involving AR glasses the research can be extended even more, making it possible to incorporate both hands in an augmented world by freeing the other hand from holding a mobile device. It will be fascinating to follow the development of these technologies.
Bibliography


Online references


Appendix - SUS-survey answers
System usability scale (SUS)

I think that I would like to use this system frequently

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
</table>
| ![Strongly disagree](https://docs.google.com/forms/d/1HqBRejV9oUDVLuroA4u9zkI2qHCLqxmvpCulIh0g/edit#response=ACYDBNh_rCSguQHZ1NxmEET4VEwzCC1FvPi...)
| ![Strongly agree](https://docs.google.com/forms/d/1HqBRejV9oUDVLuroA4u9zkI2qHCLqxmvpCulIh0g/edit#response=ACYDBNh_rCSguQHZ1NxmEET4VEwzCC1FvPi...)

I found the system unnecessarily complex

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
</table>
| ![Strongly disagree](https://docs.google.com/forms/d/1HqBRejV9oUDVLuroA4u9zkI2qHCLqxmvpCulIh0g/edit#response=ACYDBNh_rCSguQHZ1NxmEET4VEwzCC1FvPi...)
| ![Strongly agree](https://docs.google.com/forms/d/1HqBRejV9oUDVLuroA4u9zkI2qHCLqxmvpCulIh0g/edit#response=ACYDBNh_rCSguQHZ1NxmEET4VEwzCC1FvPi...)

I thought the system was easy to use

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
</table>
| ![Strongly disagree](https://docs.google.com/forms/d/1HqBRejV9oUDVLuroA4u9zkI2qHCLqxmvpCulIh0g/edit#response=ACYDBNh_rCSguQHZ1NxmEET4VEwzCC1FvPi...)
| ![Strongly agree](https://docs.google.com/forms/d/1HqBRejV9oUDVLuroA4u9zkI2qHCLqxmvpCulIh0g/edit#response=ACYDBNh_rCSguQHZ1NxmEET4VEwzCC1FvPi...)

I think that I would need the support of a technical person to be able to use this system

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
</table>
| ![Strongly disagree](https://docs.google.com/forms/d/1HqBRejV9oUDVLuroA4u9zkI2qHCLqxmvpCulIh0g/edit#response=ACYDBNh_rCSguQHZ1NxmEET4VEwzCC1FvPi...)
| ![Strongly agree](https://docs.google.com/forms/d/1HqBRejV9oUDVLuroA4u9zkI2qHCLqxmvpCulIh0g/edit#response=ACYDBNh_rCSguQHZ1NxmEET4VEwzCC1FvPi...)

I found the various functions in this system were well integrated

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
</table>
| ![Strongly disagree](https://docs.google.com/forms/d/1HqBRejV9oUDVLuroA4u9zkI2qHCLqxmvpCulIh0g/edit#response=ACYDBNh_rCSguQHZ1NxmEET4VEwzCC1FvPi...)
| ![Strongly agree](https://docs.google.com/forms/d/1HqBRejV9oUDVLuroA4u9zkI2qHCLqxmvpCulIh0g/edit#response=ACYDBNh_rCSguQHZ1NxmEET4VEwzCC1FvPi...)

https://docs.google.com/forms/d/1HqBRejV9oUDVLuroA4u9zkI2qHCLqxmvpCulIh0g/edit#response=ACYDBNh_rCSguQHZ1NxmEET4VEwzCC1FvPi...
I thought there was too much inconsistency in this system

Strongly disagree

Strongly agree

I would imagine that most people would learn to use this system very quickly

Strongly disagree

Strongly agree

I found the system very cumbersome to use

Strongly disagree

Strongly agree

I felt very confident using the system

Strongly disagree

Strongly agree

I needed to learn a lot of things before I could get going with this system

Strongly disagree

Strongly agree

Formuläret skapades på Linköpings universitet.

Google Formulär

https://docs.google.com/forms/d/1HqBReJv9oUDVLunoA4u9kzkl2qHCLqxmvpCulNfr0g/edit?response=ACYDBNh_rCSguQHZ1NxmxEET4VEzwCCIFvPi…
System usability scale (SUS)

1. I think that I would like to use this system frequently
   - 1: Strongly disagree
   - 2: Disagree
   - 3: Neutral
   - 4: Agree
   - 5: Strongly agree

2. I found the system unnecessarily complex
   - 1: Strongly disagree
   - 2: Disagree
   - 3: Neutral
   - 4: Agree
   - 5: Strongly agree

3. I thought the system was easy to use
   - 1: Strongly disagree
   - 2: Disagree
   - 3: Neutral
   - 4: Agree
   - 5: Strongly agree

4. I think that I would need the support of a technical person to be able to use this system
   - 1: Strongly disagree
   - 2: Disagree
   - 3: Neutral
   - 4: Agree
   - 5: Strongly agree

5. I found the various functions in this system were well integrated
   - 1: Strongly disagree
   - 2: Disagree
   - 3: Neutral
   - 4: Agree
   - 5: Strongly agree
I thought there was too much inconsistency in this system

1  2  3  4  5
Strongly disagree  ○  ○  ○  ○  ○  Strongly agree

I would imagine that most people would learn to use this system very quickly

1  2  3  4  5
Strongly disagree  ○  ○  ○  ○  ○  Strongly agree

I found the system very cumbersome to use

1  2  3  4  5
Strongly disagree  ○  ○  ○  ○  ○  Strongly agree

I felt very confident using the system

1  2  3  4  5
Strongly disagree  ○  ○  ○  4  ○  Strongly agree

I needed to learn a lot of things before I could get going with this system

1  2  3  4  5
Strongly disagree  ○  ○  4  ○  ○  Strongly agree

Formuläret skapades på Linköpings universitet.

Google Formulär
### System usability scale (SUS)

I think that I would like to use this system frequently

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>O</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Strongly disagree

Strongly agree

I found the system unnecessarily complex

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>O</td>
<td></td>
</tr>
</tbody>
</table>

Strongly disagree

Strongly agree

I thought the system was easy to use

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>O</td>
<td></td>
</tr>
</tbody>
</table>

Strongly disagree

Strongly agree

I think that I would need the support of a technical person to be able to use this system

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>O</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Strongly disagree

Strongly agree

I found the various functions in this system were well integrated

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>O</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Strongly disagree

Strongly agree
I thought there was too much inconsistency in this system

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>Strongly disagree</td>
<td></td>
</tr>
</tbody>
</table>

I would imagine that most people would learn to use this system very quickly

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>Strongly disagree</td>
<td></td>
</tr>
</tbody>
</table>

I found the system very cumbersome to use

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>Strongly disagree</td>
<td></td>
</tr>
</tbody>
</table>

I felt very confident using the system

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>Strongly disagree</td>
<td></td>
</tr>
</tbody>
</table>

I needed to learn a lot of things before I could get going with this system

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>Strongly disagree</td>
<td></td>
</tr>
</tbody>
</table>
System usability scale (SUS)

I think that I would like to use this system frequently

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I found the system unnecessarily complex

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I thought the system was easy to use

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I think that I would need the support of a technical person to be able to use this system

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I found the various functions in this system were well integrated

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### System usability scale (SUS)

23/08/2018

<table>
<thead>
<tr>
<th>Question</th>
<th>Likert Scale</th>
<th>Rating</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I thought there was too much inconsistency in this system</td>
<td>1 2 3 4 5</td>
<td>3</td>
<td>Strongly disagree</td>
</tr>
<tr>
<td>I would imagine that most people would learn to use this system very quickly</td>
<td>1 2 3 4 5</td>
<td>5</td>
<td>Strongly agree</td>
</tr>
<tr>
<td>I found the system very cumbersome to use</td>
<td>1 2 3 4 5</td>
<td>3</td>
<td>Strongly disagree</td>
</tr>
<tr>
<td>I felt very confident using the system</td>
<td>1 2 3 4 5</td>
<td>5</td>
<td>Strongly agree</td>
</tr>
<tr>
<td>I needed to learn a lot of things before I could get going with this system</td>
<td>1 2 3 4 5</td>
<td>5</td>
<td>Strongly agree</td>
</tr>
</tbody>
</table>

Formuläret skapades på Linköpings universitet.

Google Formulär
System usability scale (SUS)

I think that I would like to use this system frequently

1 2 3 4 5

Strongly disagree ○ ○ ○ ○ ○ Strongly agree

I found the system unnecessarily complex

1 2 3 4 5

Strongly disagree ○ ○ ○ ○ ○ Strongly agree

I thought the system was easy to use

1 2 3 4 5

Strongly disagree ○ ○ ○ ○ ○ Strongly agree

I think that I would need the support of a technical person to be able to use this system

1 2 3 4 5

Strongly disagree ○ ○ ○ ○ ○ Strongly agree

I found the various functions in this system were well integrated

1 2 3 4 5

Strongly disagree ○ ○ ○ ○ ○ Strongly agree
I thought there was too much inconsistency in this system

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I would imagine that most people would learn to use this system very quickly

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I found the system very cumbersome to use

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I felt very confident using the system

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I needed to learn a lot of things before I could get going with this system

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Formuläret skapades på Linköpings universitet.

Google Formulär
## System usability scale (SUS)

### I think that I would like to use this system frequently

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td>☐</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>☐</td>
</tr>
</tbody>
</table>

### I found the system unnecessarily complex

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td>☐</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### I thought the system was easy to use

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td>☐</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### I think that I would need the support of a technical person to be able to use this system

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td>☐</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### I found the various functions in this system were well integrated

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
I thought there was too much inconsistency in this system

1 2 3 4 5
Strongly disagree ○ ○ ○ ○ ○ Strongly agree

I would imagine that most people would learn to use this system very quickly

1 2 3 4 5
Strongly disagree ○ ○ ○ ○ ○ Strongly agree

I found the system very cumbersome to use

1 2 3 4 5
Strongly disagree ○ ○ ○ ○ ○ Strongly agree

I felt very confident using the system

1 2 3 4 5
Strongly disagree ○ ○ ○ ○ ○ Strongly agree

I needed to learn a lot of things before I could get going with this system

1 2 3 4 5
Strongly disagree ○ ○ ○ ○ ○ Strongly agree
# System usability scale (SUS)

**I think that I would like to use this system frequently**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**I found the system unnecessarily complex**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**I thought the system was easy to use**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**I think that I would need the support of a technical person to be able to use this system**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**I found the various functions in this system were well integrated**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Formuläret skapades på Linköpings universitet.

Google Formulär

I thought there was too much inconsistency in this system

Strongly disagree

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Strongly agree

I would imagine that most people would learn to use this system very quickly

Strongly disagree

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Strongly agree

I found the system very cumbersome to use

Strongly disagree

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Strongly agree

I felt very confident using the system

Strongly disagree

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Strongly agree

I needed to learn a lot of things before I could get going with this system

Strongly disagree

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Strongly agree
## System usability scale (SUS)

<table>
<thead>
<tr>
<th>Question</th>
<th>Rating</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>I think that I would like to use this system frequently</td>
<td>4</td>
<td>4.0</td>
</tr>
<tr>
<td>I found the system unnecessarily complex</td>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>I thought the system was easy to use</td>
<td>5</td>
<td>5.0</td>
</tr>
<tr>
<td>I think that I would need the support of a technical person to be able to use this system</td>
<td>4</td>
<td>4.0</td>
</tr>
<tr>
<td>I found the various functions in this system were well integrated</td>
<td>5</td>
<td>5.0</td>
</tr>
</tbody>
</table>

**Score Interpretation:**
- **1-2**: Very poor
- **2.5-3.5**: Poor
- **4-4.5**: Below average
- **5-6**: Above average
- **6.5-7.5**: Good
- **8-10**: Very good
I thought there was too much inconsistency in this system

1 2 3 4 5
Strongly disagree ○ ○ ○ ○ ○ Strongly agree

I would imagine that most people would learn to use this system very quickly

1 2 3 4 5
Strongly disagree ○ ○ ○ ○ ○ Strongly agree

I found the system very cumbersome to use

1 2 3 4 5
Strongly disagree ○ ○ ○ ○ ○ Strongly agree

I felt very confident using the system

1 2 3 4 5
Strongly disagree ○ ○ ○ ○ ○ Strongly agree

I needed to learn a lot of things before I could get going with this system

1 2 3 4 5
Strongly disagree ○ ○ ○ ○ ○ Strongly agree
System usability scale (SUS)

I think that I would like to use this system frequently

1 2 3 4 5
Strongly disagree | | | ○ | | | | Strongly agree

I found the system unnecessarily complex

1 2 3 4 5
Strongly disagree | | | ○ | | | | Strongly agree

I thought the system was easy to use

1 2 3 4 5
Strongly disagree | ○ | | | | | Strongly agree

I think that I would need the support of a technical person to be able to use this system

1 2 3 4 5
Strongly disagree | | | ○ | | | | Strongly agree

I found the various functions in this system were well integrated

1 2 3 4 5
Strongly disagree | | | | ○ | | Strongly agree
I thought there was too much inconsistency in this system

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I would imagine that most people would learn to use this system very quickly

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I found the system very cumbersome to use

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I felt very confident using the system

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I needed to learn a lot of things before I could get going with this system

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
System usability scale (SUS)

I think that I would like to use this system frequently

1  2  3  4  5
Strongly disagree ○ ○ ○ ○ ○ Strongly agree

I found the system unnecessarily complex

1  2  3  4  5
Strongly disagree ○ ○ ○ ○ ○ Strongly agree

I thought the system was easy to use

1  2  3  4  5
Strongly disagree ○ ○ ○ ○ ○ Strongly agree

I think that I would need the support of a technical person to be able to use this system

1  2  3  4  5
Strongly disagree ○ ○ ○ ○ ○ Strongly agree

I found the various functions in this system were well integrated

1  2  3  4  5
Strongly disagree ○ ○ ○ ○ ○ Strongly agree
I thought there was too much inconsistency in this system

1  2  3  4  5

Strongly disagree    Strongly agree

I would imagine that most people would learn to use this system very quickly

1  2  3  4  5

Strongly disagree    Strongly agree

I found the system very cumbersome to use

1  2  3  4  5

Strongly disagree    Strongly agree

I felt very confident using the system

1  2  3  4  5

Strongly disagree    Strongly agree

I needed to learn a lot of things before I could get going with this system

1  2  3  4  5

Strongly disagree    Strongly agree

Formuläret skapades på Linköpings universitet.

Google Formulär

https://docs.google.com/forms/d/1HqBReJv9oUDVLuroA4u9kxkI2qHCLqxmvpCulNh0g/edit#response=ACYDBNh_rCSguQHZ1NxmEET4VEwzCCIfvP
System usability scale (SUS)

I think that I would like to use this system frequently

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Strongly disagree | | | | | Strongly agree

I found the system unnecessarily complex

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Strongly disagree | | | | | Strongly agree

I thought the system was easy to use

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Strongly disagree | | | | | Strongly agree

I think that I would need the support of a technical person to be able to use this system

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Strongly disagree | | | | | Strongly agree

I found the various functions in this system were well integrated

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Strongly disagree | | | | | Strongly agree
I thought there was too much inconsistency in this system

1 2 3 4 5

Strongly disagree

|   |   |   |   |   | Strongly agree |

I would imagine that most people would learn to use this system very quickly

1 2 3 4 5

Strongly disagree

|   |   |   |   |   | Strongly agree |

I found the system very cumbersome to use

1 2 3 4 5

Strongly disagree

|   |   |   |   |   | Strongly agree |

I felt very confident using the system

1 2 3 4 5

Strongly disagree

|   |   |   |   |   | Strongly agree |

I needed to learn a lot of things before I could get going with this system

1 2 3 4 5

Strongly disagree

|   |   |   |   |   | Strongly agree |

Formuläret skapades på Linköpings universitet.

Google Formulär
System usability scale (SUS)

I think that I would like to use this system frequently

1  2  3  4  5
Strongly disagree 〇 〇 〇 〇 〇 〇 〇 〇

I found the system unnecessarily complex

1  2  3  4  5
Strongly disagree 〇 〇 〇 〇 〇 〇 〇 〇

I thought the system was easy to use

1  2  3  4  5
Strongly disagree 〇 〇 〇 〇 〇 〇 〇 〇

I think that I would need the support of a technical person to be able to use this system

1  2  3  4  5
Strongly disagree 〇 〇 〇 〇 〇 〇 〇 〇

I found the various functions in this system were well integrated

1  2  3  4  5
Strongly disagree 〇 〇 〇 〇 〇 〇 〇 〇

https://docs.google.com/forms/d/1HqBReJv9oUDVLuroA4u9kzkI2qHCL4xmvopCulNfr0g/edit?response=ACYDBNh_rCSguQHZ1NxmEET4VEzwzCCIFvP
I thought there was too much inconsistency in this system

Strongly disagree (1) 2 (2) 3 (4) 4 (5) Strongly agree

I would imagine that most people would learn to use this system very quickly

Strongly disagree (1) 2 (2) 3 (4) 4 (5) Strongly agree

I found the system very cumbersome to use

Strongly disagree (1) 2 (2) 3 (4) 4 (5) Strongly agree

I felt very confident using the system

Strongly disagree (1) 2 (2) 3 (4) 4 (5) Strongly agree

I needed to learn a lot of things before I could get going with this system

Strongly disagree (1) 2 (2) 3 (4) 4 (5) Strongly agree

Formuläret skapades på Linköpings universitet.
A.1 SUS-survey statements

I think that I would like to use this system frequently

- 0 (0 %)
- 2 (16.7 %)
- 5 (41.7 %)
- 5 (41.7 %)
- 0 (0 %)

I found the system unnecessarily complex

- 5 (41.7 %)
- 4 (33.3 %)
- 2 (16.7 %)
- 1 (8.3 %)
- 0 (0 %)

I thought the system was easy to use

- 0 (0 %)
- 2 (16.7 %)
- 2 (16.7 %)
- 7 (58.3 %)
- 1 (8.3 %)
A.1. SUS-survey statements

I think that I would need the support of a technical person to be able to use this system

![Graph showing responses to the SUS-survey statement about needing technical support.](image)

I found the various functions in this system were well integrated

![Graph showing responses to the SUS-survey statement about function integration.](image)

I thought there was too much inconsistency in this system

![Graph showing responses to the SUS-survey statement about inconsistency.](image)
A.1. SUS-survey statements

I would imagine that most people would learn to use this system very quickly

I found the system very cumbersome to use

I felt very confident using the system
I needed to learn a lot of things before I could get going with this system