NTNU
Norwegian University of
Science and Technology

An educational game for enhanced medical imaging interpretation skills

Kristoffer Andreas Dalby
Fredrik Borgen Tørnvall

Master of Science in Informatics
Submission date: June 2017
Supervisor: Frank Lindseth, IDI

Norwegian University of Science and Technology
Department of Computer Science

Abstract

In the field of medicine, medical imaging is a tool of significance. It is a noninvasive diagnostic method and an instrument for guiding surgical equipment inside the patient. There exist different types of image modalities and technologies for different areas of medicine. What they all have in common, is that in order to utilize the tools to its full potential, practice and training are required.

This thesis evaluates the current state of medical imaging interpretation learning software. The goal is to improve previous work and expand available software to further enhance how medical staff can practice medical imaging interpretation. Emphasis is put on game elements and its ability to encourage users to learn.

A software system, Medical Imaging Interpretation Courseware (MIIC), was chosen to be used as the basis for a new platform to reach the goal. MIIC implements classroom-centered training and an admin interface to help the creation of exercises. The admin interface was improved to make collecting and generating image-based questions simpler and more efficient for the administrators.

A new application[1] was created with a focus on compatibility with all platforms, especially mobile devices for easy access. The application utilizes game elements like scoring to motivate the users to continuously strive to increase their high score, and improve their skill in medical imaging interpretation.

The application was tested on a group of medical students, and the results showed that most users found the game elements motivating and perceived the application as useful. The test users reported that the application would be beneficial as a supplement to their already existing training program.

[1] `https://miip.idi.ntnu.no/app`

Sammendrag

Innen medisin er medisinske bilder et viktig verktøy. De brukes for å diagnostisere på en ikke-invasiv måte eller guide instrumenter på innsiden av en pasient. Det finnes mange forskjellige modaliteter, basert på forskjellig teknologi og som brukes til forskjellige medisinske områder. Det alle har til felles er at for å bruke dem på en effektiv måte må man trene på tolkning av bildene.

Denne oppgaven ser på tidligere forsøk på å lage programvare som skal tilrettelegge for trening på medisinsk bilde tolkning, og i tillegg utvide denne programvare. Vekten er lagt på å inkorporere spill elementer i programvaren for å motivere for læring.

Etter en evaluering ble valget å videreutvikle Medical Imaging Interpretation Courseware (MIIC) tatt. Denne løsningen er en kursplattform som er lagt til rette for å gjennomføre undervisning i forsamlinger. MIIC ble utvidet med forbedringer i arbeidsflyten til administrator hvor det ble lagt vekt på enklere oppgave og fasit generering.

Som et supplement til kursplattformen ble en ny web applikasjon[2] med fokus på mobile enheter laget. Denne applikasjonen fokuserer på læring og bruker spill elementer som motivasjonsfaktor for læring.

Applikasjonen er testet på en gruppe medisinstudenter hvor resultatet indikerer at spill elementene er motiverende. Test brukerne mener også at en slik applikasjon ville være nyttig som et supplement til deres trening.

[2] https://miip.idi.ntnu.no/app

Preface

This thesis is the result of the work completed for our Master of Science and Technology degrees in Informatics at the Norwegian University of Science and Technology. The work was carried out during the autumn semester of 2016 and spring semester of 2017.

We would like to thank our supervisor Dr. Frank Lindseth for his help and guidance during this project. We would also like to thank Emilie Amdal Nygren for proof reading and constructive feedback.

Fredrik Borgen Tørnvall and Kristoffer Andreas Breiland Dalby

Trondheim, June 8, 2017

Table of Contents

List of Tables

List of Figures

Abbreviations

HTML	=	HyperText Markup Language
API	=	Application Programming Interface
CPU	=	Central Processing Unit
CSS	=	Cascading Style Sheet
CT	=	Computed Tomography
DRF	=	Django Rest Framework
GA	=	Google Analytics
HTTP	=	Hypertext Transfer Protocol
IT	=	Information Technology
JSON	=	JavaScript Object Notation
MIIC	=	Medical Imaging Interpretation Course
MIIG	=	Medical Imaging Interpretation Game
MIIP	=	Medical Imaging Interpretation Platform
MR	=	Magnetic Resonance
MRI	=	Magnetic Resonance Imaging
OS	=	Operating System
PNG	=	Portable Network Graphics
REST	=	Representational State Transfer
RGBA	=	Red Green Blue Alpha
RWD	=	Responsive Web Design
SQL	=	Structured Query Language
SUS	=	System Usability Scale
UI	=	User Interface
URL	=	Uniform Resource Locator
US	=	Ultrasound
XML	=	Extensible Markup Language

Chapter 1

Introduction

Ultrasound is a valuable medical tool and its use in routine medical procedures has increased (1). Portability and lowered cost have made ultrasound a popular noninvasive tool for diagnostics purposes and an instrument for guiding surgical equipment inside patients. However, to fully utilize the power of ultrasound the medical staff that operates it has to be trained in interpretation of the images. When properly trained, medical staff can use ultrasound in daily routine procedures to increase patient safety and comfort (2). Portable ultrasound equipment can also aid in cases where routine procedures do not go as planned. In these cases, ultrasound is even more valuable as a tool as they potentially prevent dangerous situations for the patient.

Every year a new set of medical students start the journey towards a medical degree. Many of them will work with medical imaging in their career. It is therefore important that they get the knowledge and experience in interpreting these images. Visual inspection of medical images such as ultrasound, CT or MRI requires practise, and the opportunity to practise on existing medical images in an e-learning environment would give the students the tools to get the hours of practise they need.

1.1 Thesis goal

The goal of this thesis is to evaluate previous work and develop a serious game that could help facilitate the training of medical personal in medical imaging interpretation. With a serious game, we can increase the number of students that receive the training and lower the number of supervisors needed. The goal of a serious game, other than to entertain, is to educate, train and inform. The aim of this research is to integrate game elements in the training application to keep the trainees interested in the subject and increase their willingness to learn.

1.2 Research questions

- RQ1: How can we incentivize the use of this system?
  - Is the game fun to play?
  - Does scoring motivate the player?
- RQ2: To what extent is the application perceived as a useful learning tool?
  - Is the game perceived as a useful learning tool?
  - Does the application provide adequate challenge?
  - Does the zoom functionality help the users to be more accurate?

1.3 Thesis outline

The thesis contains eight chapters.

- **Chapter 2 – Background:**

  Introduces the background theory for the material used in this thesis.

- **Chapter 3 – Methodology:**

  Describes methodology used for development of the application.

- **Chapter 4 – Implementation:**

  Presents how the application was implemented.

- **Chapter 5 – Results:**

  Presents the results of the development and user feedback.

- **Chapter 6 – Discussion:**

  Discusses the results from the user test and answers research questions.

- **Chapter 7 – Conclusion:**

  Summarizes the project and presents the conclusion for this thesis.

- **Chapter 8 – Future work:**

  Present suggestions for future work based on feedback.

Chapter 2

Background

Developing a serious game for educational purposes requires domain knowledge. This chapter will present the background information needed to tailor an application for this purpose. The educational content of the game will focus on interpreting medical images and identifying structures. The first part of this section will focus on different types of medical imaging technologies. The second part will contain information about serious gaming, and the last part will present web-based development with a focus on mobile platforms.

2.1 Medical imaging

Medical imaging is used to create an image of the internal structure of patients. These images are often used before surgery to plan the procedure, during diagnostics and surgery. There are different image modalities that can be used to create the medical image. Which modality to be used depends on the situation and what part of the body to examine. (3)

2.1.1 Radiography

Radiography was made possible when x-rays were discovered in 1895. Radiography is performed when x-rays are passed through a patient, often in a short burst. The x-rays reach a detector on the other side of the patient, and an image is formed. Images are in grayscale and the parts of the images that is the lightest are the part of the body that has absorbed most of the radiation. Radiography is often used to diagnose broken bones and lung cancer. (3)

2.1.2 Ultrasound

Ultrasound technology is based on high-frequency sound waves. The waves are produced by a transducer in a plastic housing. There are different transducers, focusing the sound waves differently to produce different scans. The sound waves are reflected from anatomical structures that have different abilities to bounce echo. To make an image from each received echo, the scanner needs to know how long it took the echo to be received and how strong the echo was. The scanner then knows which pixels to color, thus creating the ultrasound image. (3)

2.1.3 Doppler Ultrasound

This ultrasound technology takes advantage of the Doppler effect. Doppler effect is a change in pitch of sound from a fixed point. Knowing this, it is possible to use ultrasound to detect movement in blood, both in speed and in direction. Adding color to the computer interpreted image makes it possible to inspect blood movement in the heart. (3)

2.1.4 Computer Tomography

Computer tomography (CT) scan uses a computer-processed combination of x-ray images taken from different angles. Doing this creates a cross-sectional image of the scanned area. The scan is often done around a single axis of rotation, creating a large two-dimensional image. It is then possible to generate three-dimensional images using digital geometry processing on the two-dimensional image. Combining x-ray and computer-processing produces a sharper image of soft tissue than a normal x-ray. (3)

2.1.5 Magnetic Resonance Imaging

Magnetic resonance imaging (MRI) uses magnetic fields that are about 10000 to 60000 times stronger than the earth's magnetic field, radio waves and field gradients to generate images of organs. Because different types of tissue have different local magnetic properties, images made with MRI show great contrast and is often used to scan the brain. Often compared to CT and CAT scan, MRI does not involve x-ray and is considered to be a safer alternative. MR images and ultrasound images are often combined during surgery. (3)

2.2 Image guided therapy

Image-guided therapy enables medical staff to look inside patients without surgery. In many cases this enables doctors to diagnose the patient faster and safer. It is also used as a tool during surgery by connecting CT and MR images taken before surgery. By doing this, it is possible to plan the surgery better and avoid damaging the patient during surgery.

2.3 Serious gaming

Serious games is a growing market in the video game industry and academic research. Although serious games genre has existed for many years and does not exclusively contain digital games, there is still no definitive definition of what a serious game is. However there are traits that they have in common, they are games design not only for entertainment but also for a secondary purpose such as education, advertising, art or simulations. According to Valerie J. Shute, Lloyd Rieber, & Richard Van Eck (4), there are elements that have to be included in an educational game; challenge, rules, goals, feedback, interaction and representation. In addition to these features, games that promote learning needs to engage the player and induce the flow state where the education benefit is the highest (5). To achieve this state, the challenges has to be optimized to avoid anxiety and boredom.

For the purpose of this paper, the focus will be on educational games and game elements that can be added to an application to promote learning and motivate players to learn.

Anyone who makes a distinction between games and learning doesn't know the first thing about either.
— Marshall McLuhan

2.4 Games and learning

When considering games as a learning tool it is important to look at game elements that improve the motivation to learn, and to evaluate the retention of the information acquired by the user after playing instructional games. According to Garris (2002) there has been a shift in how users learn. The traditional didactic model, which focuses on instructional learning or "learning by listening" is going away in favour of a model with a more active learner role, where students learn by doing (6).

By incorporating games into the learning process enables the students to be more active. There has been studies that show that games can be used as a tool to improve the learning of advanced subjects (7) (8). A literature review by Whitehill (1992) found that implementing game elements into learning improves the learning benefit (9).

Another important aspect of using game elements is to provoke the user's motivation. One of the core hallmarks of a game is the goal of getting a player to play over and over again, pulling the user back into another game when they finish (6). To replicate this hallmark and motivate the player to repeat the learning process would be a goal. A study finds that games in education seem to effectively motivate the students and make them more interested in the subject matter, but how well it can be interpreted as effective learning is less clear (10).

In a scenario where the goal is to learn medical imaging interpretation, the motivation for learning is mostly professional. In other words the motivation is extrinsic, where the outcome is the motivation for success. An application design to teach should strive to make the user interested in the actual task at hand, triggering the intrinsic motivation (11). The goal of the game should be to motivate the players to become interested in both the content of the game and the outcome of playing it, while the outcome should be impotent in the real-world (6). When considering motivation and success, Locke (1990) indicates that specific and difficult goals lead to improved performance (12).

> *If video games can be transformed so that their users learn, a great many people may come to understand and control dynamic systems*
>
> — Simons (1993) (13)

There exist a wide array of different games; all utilizes different characteristics making them different from each other. There are several suggestions on how these can be categorized and what features are the core characteristic. Based on their meta-study, Garris (2002) suggests six groups of characteristics to categorize most games; fantasy, rules/-goals, sensory stimuli, challenge, mystery, and control (6). Most games would fit into one or more of these categories, and incorporating some of these features would possibly yield good results. Thomas (1994) suggests that an important core characteristic of games is that they have no real-world consequences. In the field of medical training, this may be an important characteristic as it allows for users to learn without any risk, building confidence in a safe environment (14).

2.5 Related work

There have previously been attempts at creating training applications for ultrasound interpretation. In 2015 a mobile application prototype was developed with the goal of teaching ultrasound-guided regional anesthesia (15). After analyzing the prototype and concluding that a more general approach was needed so that the application could be used in other areas of medicine, two other projects was started. One project focused on teaching medical imaging in a course setting (MIIC) with the ability to create questions and lectures (16). The other project focused on generalizing the prototype from 2015, re-implementing it with web technology (17). Both projects featured assignments with different images modalities and various medical topics.

A tool for solution creation was also an area that needed focus. This feature was missing from the prototype created in 2015. The thesis concluded that a solution creation tool was needed, and that it should collect the answer to a question from more than one expert. It should also be possible to get an average solution calculated from those solutions.

2.5.1 Medical Imaging Interpretation Game

Medical Imaging Interpretation Game (MIIG) (17) is a project that was started as a master theses, and the goal was to build an application based on the findings of the 2015 project. The application was divided into two solutions, one game, the part of the project the users will interact with, and an administrator interface. Based on the finding from the 2015 prototype the initial goal was to develop a native mobile application. However, after some research it was decided that a web application would be more beneficial. The reason being that it would be easier to maintain and that the application could be made available on more that one platform. Another benefit was also the ability to update on demand, without going trough any content provider's store for approval.

2.5.2 Medical Imaging Interpretation Course

Medical Imaging Interpretation Course (MIIC) (16) is the project that was conducted at the same time as MIIG, focusing on organizing and implementing courses on medical imaging. The MIIC project focused on courses held for a big audience with the participants working on computers. The solution is accessible on mobile devices, but it is not designed for them. This project was also implemented as a web application with a course mode for the participants and an administration interface for the course administrators.

2.6 Existing solutions

In this section, three existing solutions will be looked at. The criteria for these applications is that the game, training application or mobile application is software designed to train personnel in interpreting medical imaging.

2.6.1 A Serious Game for Learning Ultrasound-Guided Needle Placement Skills

The first solution is a serious game prototype made to teach ultrasound-guided needle placement, integrating game elements for active and efficient learning. The game focuses on procedures such as biopsy, local anesthesia, and fluid drainage, which are necessary skills in minimal invasive surgery (MIS) (18).

This solution took on the simulation approach, where the player interacts with 3D objects in a 3D world. The game's goal is to simulate the needle placement and employ haptic feedback to increase realism. Game elements are implemented in training to increase willingness to learn, such as time trials, hints, performance evaluation and scoring feedback (18).

2.6.2 WeBSurg

WeBSurg is e-learning website offering educational content in the field of minimally invasive surgery. This is a web-based solution which offers videos, lectures, and articles about the latest surgical breakthroughs. They also offer the opportunity to chat with surgeons and experts (19).

2.6.3 SonoAccess 2.0 Mobile App

This is a mobile training application with ultrasound resources. In this application, the user has access to video content, case studies, images and reference guides. The user can filter the content bases on specialization so that content that is not interesting for the user is hidden. This application is not a game, but rather a resource point for medical professionals interested in educational material for ultrasound (20).

Chapter 3

Methodology

In this chapter, the methodology used for the development of the application is described.

3.1 Platform

In this section, a comparison of the different available development platforms will be given. First a short entry about the earlier work, and what platforms they incorporate. The following sections will discuss the pros and cons of the different available platforms, Native, Cross-platform and Web based. At the end, an evaluation and the selected platform will be presented.

3.1.1 Continuation of earlier work

Before commencing development of a new system, earlier work was considered. Regarding the given domain, there were two systems to consider, MIIG and MIIC. Both systems are web based applications built as backend services, rendering everything on a server. As both applications are built with desktop and bigger screens in mind, they have not designed the user interface for smaller screens. Taking this into consideration they did not fit the requirements for a web application with a native mobile application feel.

A decision was made to choose one of the systems as the backend application for the new artifact. Choosing one of the existing systems gave the possibility to not re-implement the administrator interface in addition to the data management model already developed. After evaluating both systems, MIIC was chosen. MIIC was selected based on wishes from the administrators, and it was found to be the most sophisticated system of the two. In addition, knowledge of the technology used to build MIIC was greater within the team of developers.

3.1.2 Native and Cross-platform

A native application is software written and compiled to low-level machine code for a particular platform that runs directly on either a virtual machine or the CPU. Writing native application may give access to further optimization and platform specific features, where controlling underlying hardware may be one of them (21). The performance for native applications might also be significantly better than other solutions (22). Dependent on the platform, there may be multiple languages or frameworks that can be used to create native applications (23). Using Apple iOS platform as an example, using either Apple's Swift[1] or Objective-C[2] programming language will allow the programmer to write native applications for the given platform. The drawback, however, is that the application cannot be easily converted to run on a competing platform like Android[3]. There are also some security concerns when executing code directly on a device, but each vendor has strict security measures put in place to mitigate these risks (24).

Using a cross-platform solution, for example, Xamarin[4] will allow a programmer to share a significant amount of the code across multiple platforms and compile native applications for each platform (25). Even though a cross-platform solution is used, the different platforms may require the developer to write the user interface code for each platform to ensure that the application provides an interface that follows the design guidelines (26) of the platform.

[1] https://developer.apple.com/swift/
[2] https://en.wikipedia.org/wiki/Objective-C
[3] https://www.android.com
[4] https://www.xamarin.com

Both platform-specific and cross-platform native applications have the advantage of greater performance and possibly better integration with the particular platform's user interface guidelines. Drawbacks when developing native applications are challenges of distributing the complete application, as it must be packaged and sent out to each device. Typically through the given platforms digital distribution platform, for example, App Store[5] on Apple devices and Google Play[6] on Android devices.

3.1.3 Web based

In contrast to native applications, web applications are delivered on demand to a device by downloading the application from a web server. Web applications are typically written in an interpreted programming language (JavaScript) (27), where the entire application is downloaded from a web server and then run on the given platform. As the application is mainly executed through the system's web browser, there are limitations regarding which platform specific features the application can access. Typical limitations are access to the hardware and storage of the device.

A typical web application is written using standardized languages. The structure of a web application and a web page is Hyper Text Markup Language (HTML) (28), and the styling is Cascade Style Sheet (CSS) (29). The program code for executing logic and handling interactivity is JavaScript.

A web application can very easily be redistributed and updated as the user will only need to refresh the web page to load the new version of the software. The performance of the application is mostly bound to both the speed of the web browser and the code of the application itself. The application cannot run faster than the web browser can interpret and execute the code, and code written poorly will also affect the performance.

[5] `https://www.appstore.com/`
[6] `https://play.google.com/`

3.1.4 Evaluation

When considering the platform to be used for this thesis, it was important to discuss the different aspects of the project. The solution should work on as many platforms as possible, with a mobile-first strategy to give users the opportunity to use the application on multiple devices for training their skill. The application should have an efficient release cycle allowing for rapid deployment to the end users.

Given the properties of web-based applications (30), this strategy was chosen. Developing a web application would only require the development of one application, which can be used on all platforms with a web browser. The web-based approach also allows for easy and efficient releases to the users to get faster feedback.

Using native applications would require more platform specific code, and most likely meant that multiple applications would have to be created. When deploying the application to mobile platforms, the application would also have to go through approval processes for each platforms distribution network. Using a native platform solution to develop this project would, therefore, be too time-consuming and a web-based application is chosen.

3.2 User interface

The user interface is where the user interacts with the application, and it is the main factor in making an application accessible. Presented in this section is the users and their devices, an introduction to UI design, and an introduction to responsive design and mobile-first development.

3.2.1 Devices

To understand how the users access information today one must remember that most of them has access to mobile technology such as smartphones and tablets. A good measure to determine how they use that technology is to analyze how the user access web pages. Considering that a lot of information, communication, and knowledge is shared via the world wide web, most people in the segment for a learning application is used to acquiring information on the web.

To get a clear picture of device usage and technology available, statistics was collected from *StatCounter* (31) via their GlobalStats database. StatCounter is an independent statistics collector basing their statistics on data from 15 billion page views per month recorded across more than 2.5 million websites (32). All statistics span from August 2015 to May 2017.

The first data presented in Figure 3.1 shows the latest rise of mobile devices, smartphones. In October 2016, the number of mobile devices online surpasses desktop clients for the first time and the trend does not seem to end there. Considering the vast majority of citizens in developing countries getting access to the internet via the emerging growth in smartphone outreach (33), this is no surprise.

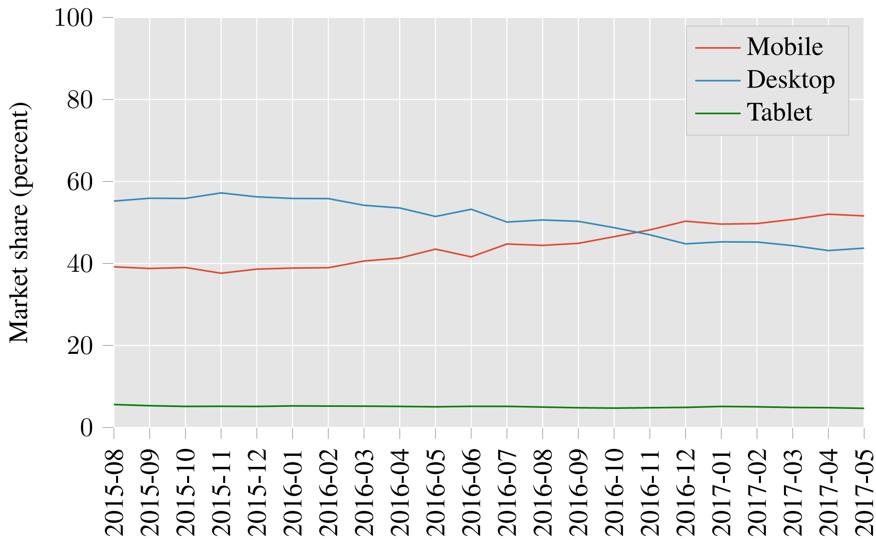


Figure 3.1:
Market share of desktop, tablet and mobile devices

In addition to market share of the device type, considering what vendor and platform used is also important. Presented in Figure 3.2 is an overview of the different vendors market share. Looking at Figure 3.2 and Figure 3.3, one can see tendencies that most smart devices run either Google Android[7] or Apple iOS[8] operating system. Using this information indicates that supporting these devices and the most popular browsers and screen resolutions will be a reasonable choice. Adding Microsofts Internet Explorer Mobile to the list, even though the market share is small, will cover most devices.

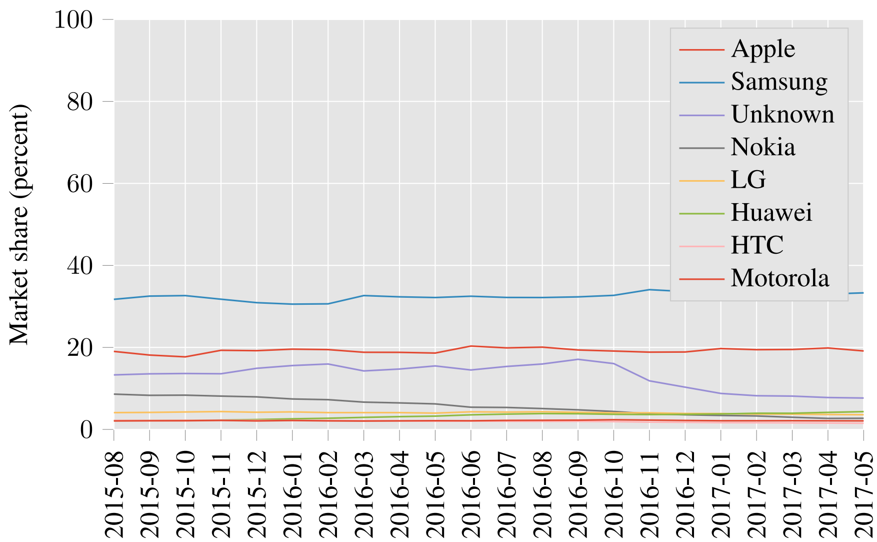


Figure 3.2:
Vendor share of the mobile market

[7] https://www.android.com
[8] https://www.apple.com/ios/

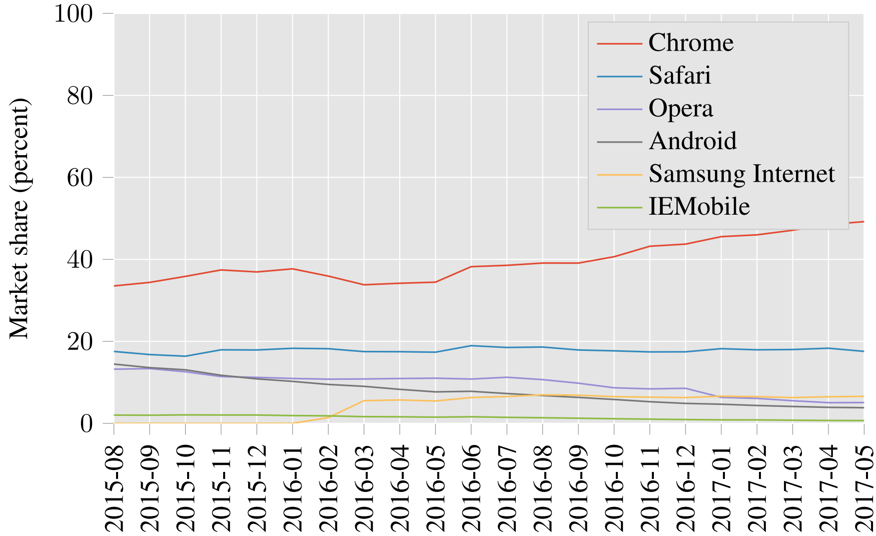


Figure 3.3:
Mobile web browser market share

Data and concerns regarding desktop browsers have been omitted as the same vendors create browsers for the mobile, tablet and desktop platforms. Each browser developer reuses the code across platforms making every feature available in the mobile browser also reachable on the tablet and desktop.

3.2.2 Design of user interfaces

The user interface (UI) is a bridge between the application and the user. All interaction happens with the UI and therefore a good interface is the key to a successful application. If a UI is poorly designed, users might struggle with the application and in extreme circumstances reject it. The UI should give the user confidence in the application.

A good user interface can be measured by its usability:

> *The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments.*
>
> — ISO 9241 part 11 (34)

The effectiveness of the UI is determined by how accurate a user can achieve a given goal in the given environment. The efficiency is determined by how much resources the user needs to achieve the specific goal in the environment. The satisfaction is measured by how much comfort and acceptability the users get by using the system. (34)

Effectiveness

To providing a UI with high effectiveness, it is important to organize what the user sees on the screen in a logical way. It is important to be consistent in terms of elements on the screen: choices of colors, the size of elements, consistency in font usage and font size and how the different elements are grouped together (35). Jakob Nielsen provides a set of commonly referred to and used guidelines for creating user interfaces with good usability (36). These guidelines suggest that the application should focus on making the user interface easier to understand and minimize the possibility for errors to occur. One way to ensure this is to make the interface reflect the users “real world”, giving the users a head start and making them feel more at home. Focusing on recognition rather than recall allows the user to interact with the UI with the information needed on the screen. Meaning the user does not have to remember how to use the application, the application explains to the user how to use it. Another important focus of good user interface design is to give the user the ability to get out of an unwanted state. The user should always be able to see the state of the application and receive frequent feedback.

Efficiency

Using the UI should not require a lot of the user. According to Jacob Nielsen, the design should be visually pleasing and minimalistic, only presenting the information needed to the user. When designing user interfaces, considering what information is presented to the user is important. However, it is also important to consider how the information is presented. When designing minimalistic interfaces, the presentation of elements and information should be of utmost importance. In the 1920s, German psychologists developed the Gestalt Principles (37). These are theories about how people view elements and objects as part of groups. These theories have since been adopted when considering user interface design to make designers consider how humans perceive what they see. The Gestalt Principles focuses on similarity, continuation, closure, proximity, symmetry and order. By leveraging these concepts, one can take advantage and consider how the user will interpret the given design.

3.2.3 Responsive web design

Designing an interface for multiple platforms and screen sizes is not trivial. The amount of information that can be displayed on each platform may differ drastically, both in height and width. One approach to achieving a consistent design across all devices and screen sizes is responsive web design (RWD) (38). The idea behind RWD is to create a layout that changes the size of the elements and displayed information, as the size of the viewport changes. An important factor to remember is that the value of what is shown at each screen size must be heavily considered as all information cannot necessarily be displayed on the smaller space.

3.2.4 Mobile-first development

A technique to improve the design process of RWD is to prioritize the mobile devices, the devices with the smallest screen sizes and resolutions first.

> *If you design mobile first, you create agreement on what matters most. You can then apply the same rationale to the desktop/ laptop version of the web site. We agreed that this was the most important set of features and content for our customers and business, why should that change with more screen space?*
>
> — Ethan Marcotte, *RESPONSIVE WEB DESIGN*

Mobile-first development gives the designers most constraints at the start of the process and forces the selections of the most important parts of the application for these devices. As the screen grows, one can simply scale up the design or add more information and elements that are less important to the overall experience. Mobile-first is a design philosophy that sets focus on the fact that the mobile user should not suffer because of their device, but rather make all information available to all devices. (38)

3.3 Technology

In this section, the chosen technology will be presented. First, the components of the earlier work will be discussed followed by technology extension to this solution. In the end, the selected language and frameworks for the Medical Imaging Interpretation Platform (MIIP) will be introduced.

3.3.1 Medical Imaging Interpretation Courseware (MIIC)

The backend for the new web application solution is built on the prior work by Marte Hallan called MIIC. This software is written in the Python programming language (39) and utilizes the Django web framework[9]. For styling the front-end and making the web more interactive, the Bootstrap styling framework [10] and jQuery[11] Javascript framework are used. The Django framework offers functionality for rendering Python data structures as HTML pages. In addition, Django handles all the database transaction and mapping of data from SQL (40) to Python Objects. The index page of MIIC can be seen in Figure 3.4.

3.3.2 Django REST

By default, existing Django applications do not offer any external endpoints for representing information in a well-supported structure for data exchange between applications. The MIIC application where therefore extended with a Django extension called Django REST Framework[12] (DRF). DRF adds functionality to Django for easily serializing data models defined for the application to popular and well-supported formats such as XML (41) or JSON (42).

[9] `https://www.djangoproject.com`
[10] `http://getbootstrap.com`
[11] `http://jquery.com`
[12] `http://www.django-rest-framework.org`

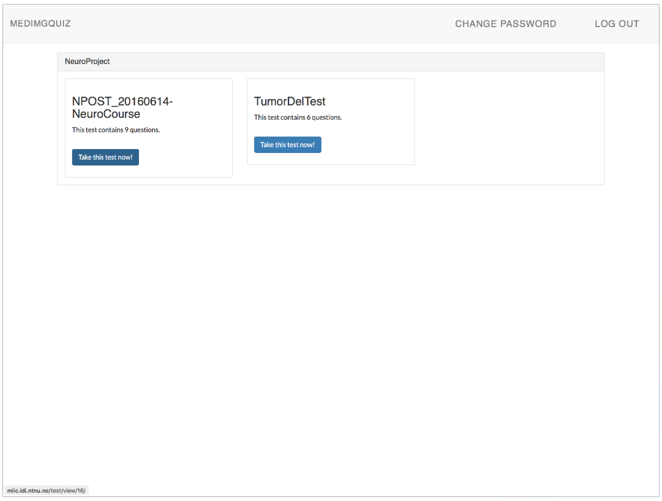


Figure 3.4:
Main view of MIIC application

By using DRF, it is possible to reuse existing code and new code which is added to the project is kept in separate files. This ensures that existing functionality is not affected by the new code. Adding an API in this way enables third party applications to use the data from the original system.

3.3.3 Elm

Elm (43) is a domain-specific programming language with functional properties created for constructing web applications. The language uses a compiler to convert Elm code to HTML, CSS, and JavaScript which can be run in a web browser.

The language is influenced and inspired by ML-based languages (44) similar to OCaml (45) and Haskell (46). It features functional programming constructs like immutable data structures[13], anonymous functions[14], and pure functions[15]. The language is entirely interoperable with HTML, CSS, and JavaScript and can be extended with features from these technologies.

[13] https://en.wikipedia.org/wiki/Immutable_object
[14] https://en.wikipedia.org/wiki/Anonymous_function
[15] https://en.wikipedia.org/wiki/Pure_function

Elm has a strong type system[16] which is enforced by the compiler in a very strict manner to guarantee virtually no runtime errors[17]. Defining the types is optional, which provides very well structured code where only ambiguous variables and functions have defined types for documentation purposes. Types are worked out by the compiler using type inference[18]. Type inference deduces the remaining types by setting up constraints on the variables or functions without known types by looking at what is known at the current point.

In Listing 3.1 a function *add* is defined taking argument *x* and *y* and executing an addition on the two arguments. As shown, there are no defined types, but the Elm runtime figures out that this is a function taking two numbers and returning one. The types are inferred by looking at the operation executed. By checking what types can be used with the + operator the program can understand what input this function can handle. In the next few lines, a simple addition with two numbers is executed, and the return value is a new number. In the third execution, a string is used as the first argument, causing a type mismatch as this is not allowed. The error message shown in Figure 3.1 is an excellent example of Elm's well-written error messages.

3.3.4 Docker

For application development and deployment, Docker[19] is used to guarantee the same environment across development and production platforms. When used in development, the Django and Elm applications are executed within containers with the same environment as the production server to minimize the possibility of incompatibilities. In production, the whole application is packaged as one container with everything the application needs to run. Packaging the application like this guarantees the same stability and runtime on every platform.

[16] https://en.wikipedia.org/wiki/Strong_and_weak_typing
[17] https://guide.elm-lang.org/error_handling/
[18] https://en.wikipedia.org/wiki/Type_inference
[19] https://www.docker.com

```
> add x y = x + y
<function> : number -> number -> number
> add 1 1
2 : number
> add "hello" 2
-- TYPE MISMATCH --------------------------- repl-temp-000.elm

The 1st argument to function `add` is causing a mismatch.
4|   add "hello" 2
         ^^^^^^^
Function `add` is expecting the 1st argument to be:
    number

But it is:
    String
```

Listing 3.1: Type inference

3.4 Architecture

The software is structured in a modified three-tier architecture (47). The application stack is composed of three layers, presentation, logic, and data layer. The logic layer is split between the back-end and the front-end part of the application.

3.4.1 Client-Server

A three-tier architecture is a client-server software architectural pattern (47) which separates the different functional layers into the various platforms of the application.

Beginning at the bottom, the data layer is located on the back-end handling the transactions of data between the database or file system. This functionality is implemented and processed through Django's Object-relation-mapper (48) which stores persistent records in a PostgreSQL[20] database and media files on the filesystem.

The logic layer is then split in two. The first part is within the Django application and handles the administrative interface implemented in MIIC, and REST API endpoints (49)

[20]https://www.postgresql.org

provides data for the training application. The second part of the logic layer resides in the front-end part of the application. The front-end implements the logic for scoring and validating the different game modes.

All functionality and logic that is needed for the application to run standalone are put into the frontend. This means that the application can quickly be repackaged as a standalone app that can function similar to a distributed native application. The logic is implemented using the Elm language.

The presentation layer is responsible for rendering the data provided from the logic layer. It is implemented in Elm and is a series of functions that transforms data structures into HTML, Javascript, and CSS.

3.4.2 The Elm Architecture

An Elm application is written in a programming pattern called The Elm Architecture[21]. The architecture describes a simple, yet straightforward way of representing the code and data your program will possess. The Elm Architecture contains the following; Model, View, Update and Runtime.

The model is a representation of your data, the state of the application, and it defines the way it is structured and what type of data it contains. Because of Elm's strong type system, the model cannot be violated, and data not defined in the model cannot exist in the application. In the following example, Listing 3.2, a simple model with only one question, is presented. The model defines a question as a record with three strings, a question, a correct answer and a wrong answer. Each part of the record has specified a type and may not be null. Below the definition of the model, an example of an initiated question is shown.

The view is a declarative[22] function which takes in the model and outputs HTML, CSS, and JavaScript. The view function can be split up into smaller functions which are responsible

[21] https://guide.elm-lang.org/architecture/
[22] https://en.wikipedia.org/wiki/Declarative_programming

```
type alias Model =
    { question: String
    , correct: String
    , wrong: String
    }

initialModel : Model
initialModel =
    { question = "Where does Winnie-the-Pooh live?"
    , correct = "Hundred Acre Wood"
    , wrong =  "Trondheim"
    }
```

Listing 3.2: Model example

```
view : Model -> Html msg
view model =
  div []
    [ h1 [] [text model.question]
    , input [type_ "radio"] []
    , text model.correct
    , br [] []
    , input [type_ "radio"] []
    , text model.wrong
    ]
```

Listing 3.3: Declarative view function

for rendering different sections of the defined model. In Listing 3.3 a view function for transforming the model in Listing 3.2 to HTML is shown. The view function is declarative and does not have side effects and will always render the same HTML given the same input.

The creation of HTML elements is performed through function calls. All functions are type checked and validated before creation. In Listing 3.4 the resulting HTML is presented.

```
<div>
    <h1>Where does Winnie-the-Pooh live?</h1>
    <input type="radio">Hundred Acre Wood
    <br>
    <input type="radio">Trondheim
</div>
```

Listing 3.4: Rendered HTML

The update part of the architecture is how the state gets modified. The state, which is the instance of the defined model with the current data, is immutable. An immutable state entails that to make changes, the entire state is replaced with a new one. The update function takes instructions and data on how to execute the modification and replaces the model with a new model. In Listing 3.5, an example of a new question with an update function is shown. The "Instruction", called Msg has one option "ChangeQuestion", which takes the new model, and replaces the old one. The update function is the *only* place where the model can be transformed.

The runtime is the underlying system that ties the three functions together. The Elm Architecture[23] has a unidirectional flow where all events and data only travels one way as shown in Figure 3.5. Given this constraint, it is important to remember that the view that is rendered will *always* be the representation of the current state of the application, and there will be no exceptions. If a user interacts with the application in any way, the runtime will send a message to the update function with the instructions needed to modify the state. When the state is changed, the new model will pass through the runtime and call the view function to update what the user sees. The next interaction will follow the same flow, and there cannot be any violations where for example the model is updated through the view function. Since the state will always follow the strict type definitions of the model, each

[23] `http://toreto.re/tea/`

view will always be reliably rendered, and each update of the state will follow a set of rules making it close to impossible to have runtime errors in the program. (43)

```
secondModel : Model
secondModel =
    { question = "When did Frozen come out?"
    , correct = "2013"
    , wrong =  "2011"
    }

type Msg =
  ChangeQuestion Model

update : Msg -> Model -> Model
update msg model =
  case msg of
    ChangeQuestion model ->
      model
```

Listing 3.5: Update function

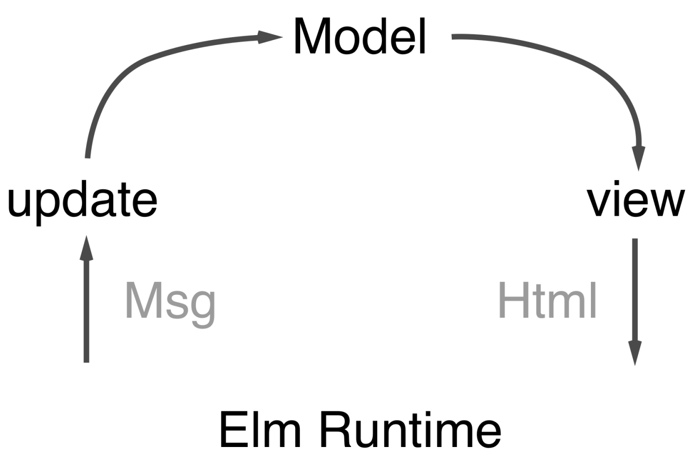


Figure 3.5: The Elm Architecture.

Chapter 4

Implementation

In this chapter, the implementation details regarding the platform (MIIP), the serious game and contributions to earlier work will be presented.

4.1 Contributions to earlier work

This section will contain contributions to earlier work.

4.1.1 REST API endpoints

To further enhance the usability of the MIIC system as a backend, a REST API layer is implemented using Django Rest Framework as presented in Section 3.3.2. Appending a layer for communicating data between the system in a programming language and platform agnostic manner is key for reusing the data outside the current platform. JSON was chosen as the data-interchange format because of its ease of use and human-readability.

In Listing 4.1 data from the Landmark module is presented in JSON format. The data is available by sending an *HTTP GET* request to the URL endpoint */api/quiz/landmarkquestion/*.

```
[
    {
        "pk": 59,
        "question": "02_MR-FLAIR_Axial",
        "original_image": "/media/Skjermbilde_2016-06-25_kl._11.34.57.png",
        "landmark_drawing": "/media/solution-Skjermbilde_2016-06-25_kl._11.34.57.png",
        "landmark_regions": [
            {
                "color": "#c02f1d",
                "name": "Tumor"
            }
        ]
    }
]
```

Listing 4.1: Landmark JSON data-interchange format

The REST API is the core data source for the new web application. However, it is important to mention that JSON is not limited to web applications, but can be used for both native Android and iOS applications, as well as for implementations on desktops.

4.1.2 Generalizing user interface

The most important part of the MIIC application, with regards to the new platform, was the administrative interface. The administrative interface purpose was to manage the different tasks, collect solution suggestions from experts and create new exercises.

In the original interface, creating a new outline or landmark question from an image required the administrator to upload the image twice, one for the outline and one for the landmark exercise. As shown in Figure 4.1, the original interface has separate menus for creating outline and landmark. There is no way to get an overview of all the tasks with the image attached, and the administrator has to remember the name of all exercises created to be able to view them or check if they are already added.

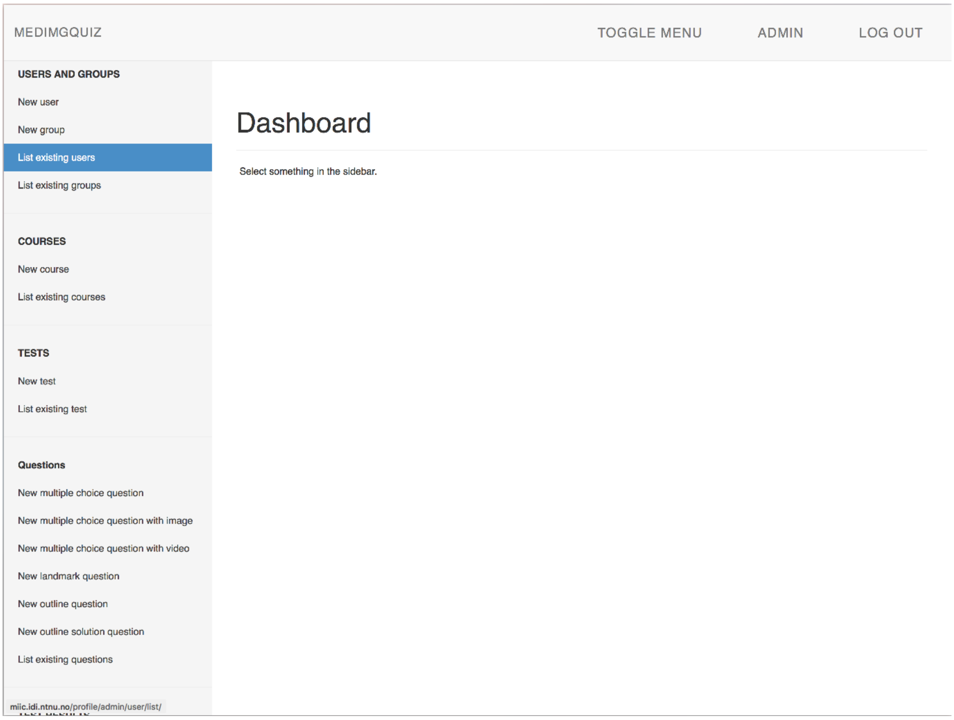


Figure 4.1: Index view of MIIC's administrative interface

When redesigning the flow for uploading images and creating better overviews, the menu was redesigned as shown in Figure 4.2. The main difference here, as will be shown in the next section, is that the outline and landmark creation has been removed in favor of *New Image* and *List existing images*. These new menu entries lead the administrator to new functionality discussed in the next section.

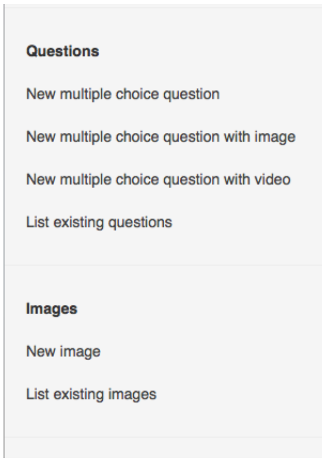


Figure 4.2: Generalized menu for handling images in MIIP

List existing images

To give the administrator an easy summary of the repertoire of images in the system, an image overview was introduced as shown in Figure 4.3. Each image is shown together with its title and gives the option to create a new landmark or outline exercise, as most pictures are suitable for both types of exercises. These images are used to collect solutions from experts, as will be discussed in section 4.1.3.

New image

Uploading images was improved with automatically detection of title and modality from an image file's name, making the upload of a lot of images more efficient. In Figure 4.4 one can see the fields auto-filled by the uploaded image name. Each text field can be overridden by the administrator.

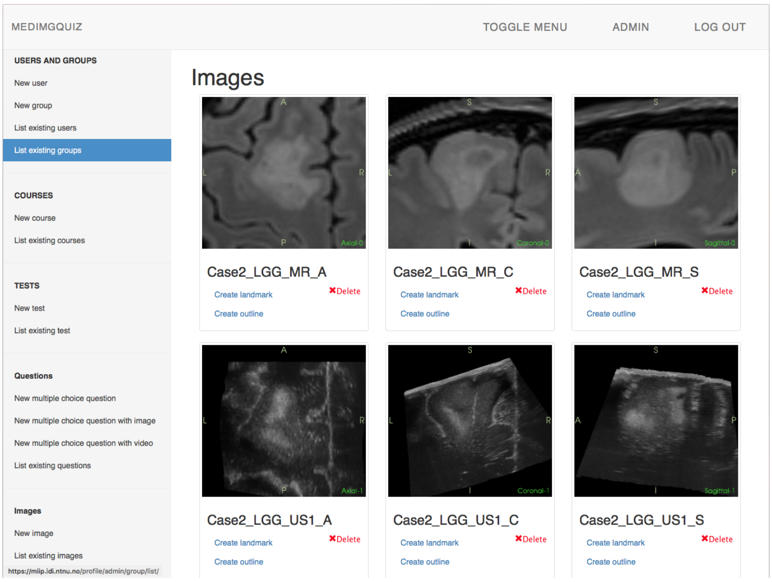


Figure 4.3: Overview of images in MIIP

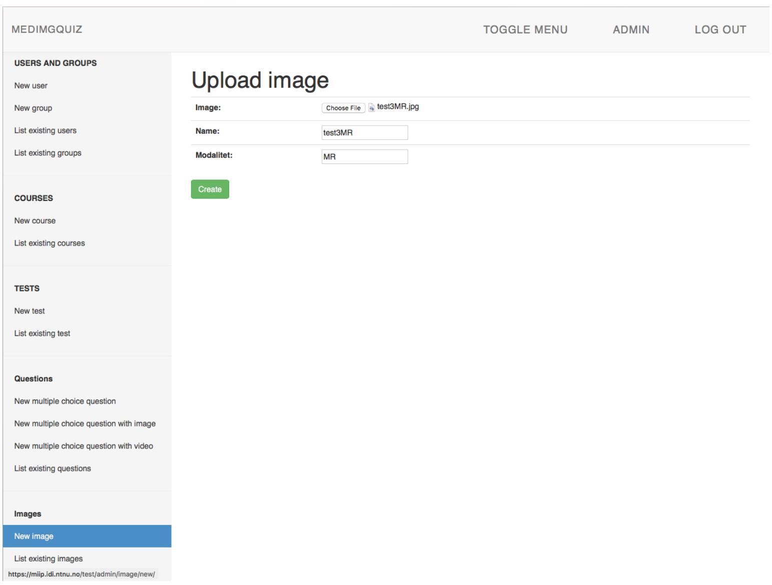


Figure 4.4: Image upload with modality detection in MIIP

4.1.3 Gathering expert suggestions

To support administrators in the work of acquiring suggestions for solutions from experts, features for creating collections of exercises without solutions and requesting suggestions were implemented. A view where the expert can input their suggestions to a set of images is created as shown in Figure 4.5.

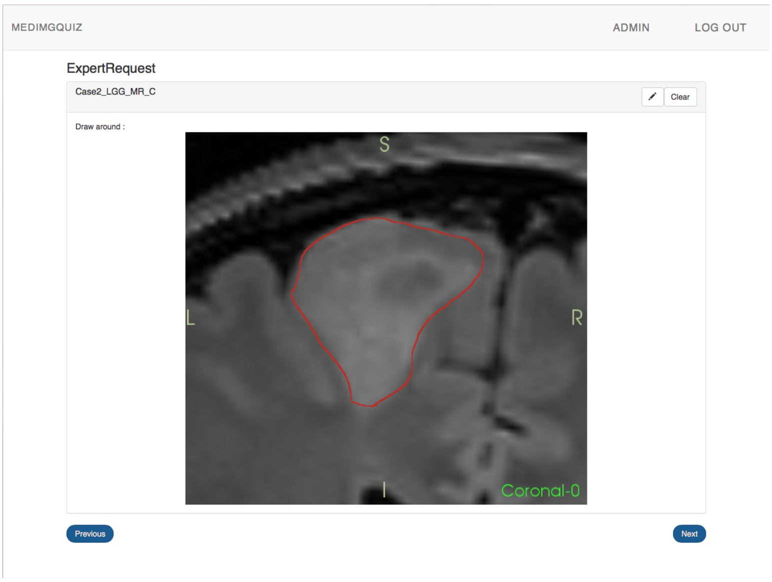


Figure 4.5: Example of an expert submission in MIIP

For the administrator, a new workflow for requesting suggestions from experts were developed. The administrator would earlier upload a picture and "draw" a solution together with the experts in a meeting, taking all discussions and contributions as one process for each picture. In the new model, the experts receive a request to record their suggestion in advance of the meeting.

To create a collection of images for experts to review, a test is created. The test is populated with "Image suggestions", which is images that have not yet been reviewed. An administrator adds images to the test either by selecting "upload image" as shown in Figure 4.4 or by selecting from images that are already uploaded, but without a solution as shown in 4.6.

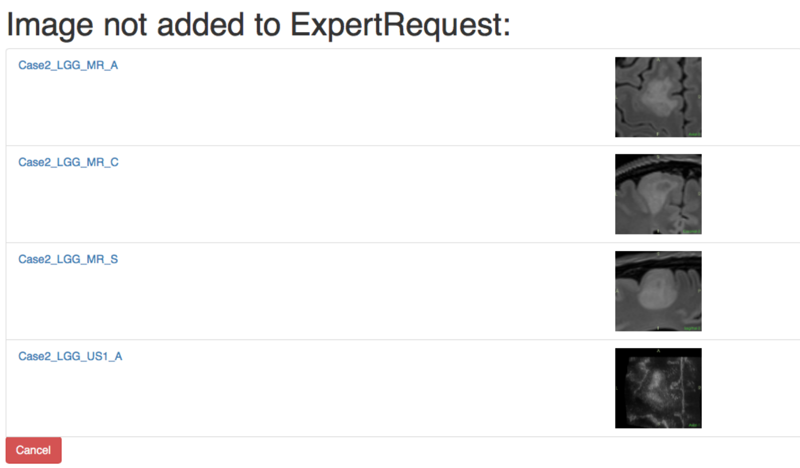


Figure 4.6: View for adding a new image to the test

Requesting suggestion via email

To further improve the solution gathering, an email feature was implemented. In Figure 4.7 the administrative overview for a request with four images is shown. The view offers information about the images added to the test, as well as a *Send email to users* button which automatically will send a notification by email to all experts registered within the working group the test belongs to. A typical working group is a collection of doctors or specialists in the field related to the different technologies or medical areas which the images is from.

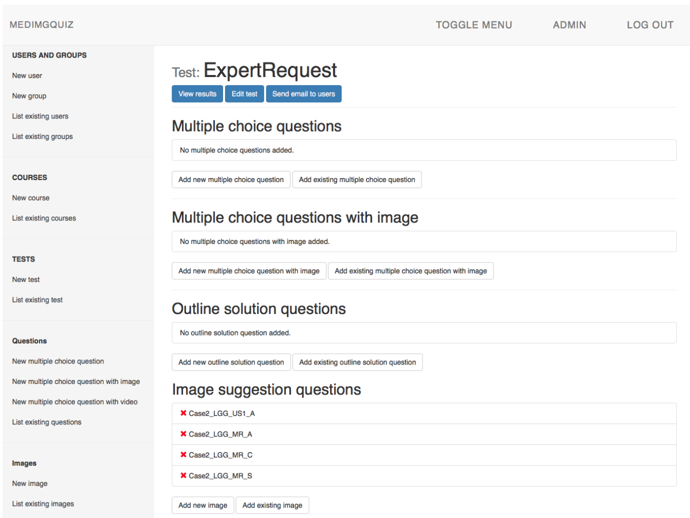


Figure 4.7: Administrative overview of expert collection test in MIIP

Calculating the average of expert suggestions

When a set of suggestions has been submitted to the system, the discussion and generation of solutions can start. In the improved system, the possibility to calculate an average of multiple solutions was introduced. The average outline is generated based on the selected expert's suggestions, where two or more can be used for the calculation. With this feature, the experts can choose to go with one of the suggested solutions or create a compromise based on several opinions. In Figure 4.8 one can see a scenario where four submitted answers are shown, where three of them are selected to generate an average solution.

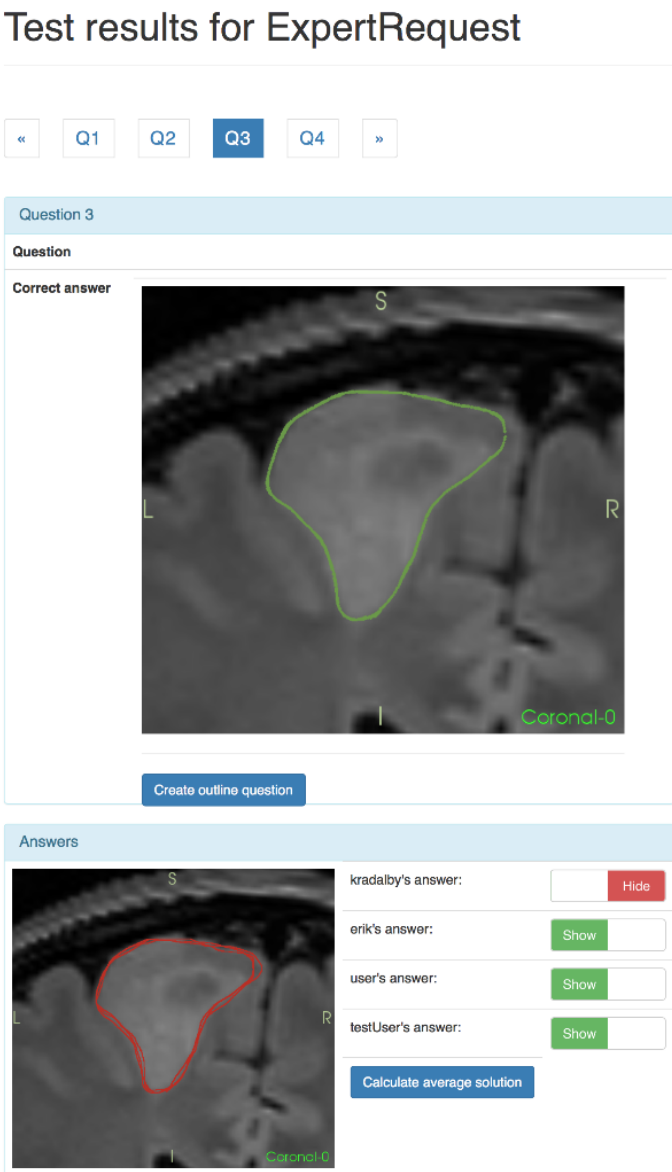


Figure 4.8: Calculating average solution based on three suggestions

Implementation

Generating an average image from two or more images is accomplished by calculating the average between two images at the time. One image is set as a reference in comparison to a second image to calculate the average. Each pair of images will be used to create a new solution based on the distance between its two outlines. To find the distance between two outlines the Euclidean distance (4.9) is calculated from each pixel in the base outline to the pixels in another outline. When a pixel pair is found, a new pixel is created in the middle between them on the output image as shown in Figure 4.10.

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}$$

Figure 4.9: Euclidean distance formula

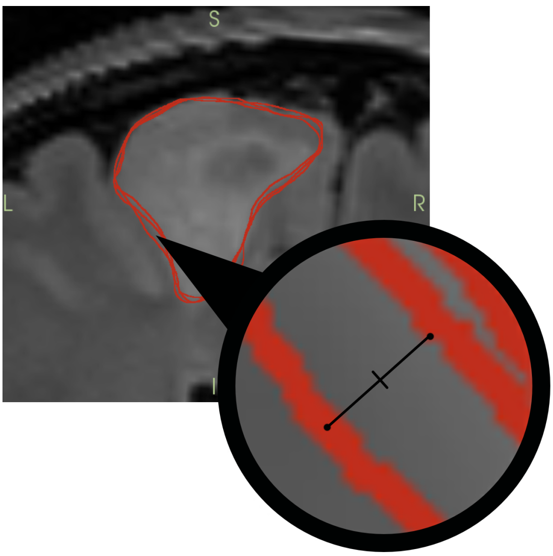


Figure 4.10: Determining the middle pixel from two outlines

When this process is finished for every pixel pair, the new output image is saved as the average of the two original images. If the average is to be calculated from more than two pictures, this process is repeated for each new picture using the average produced in the past iteration as the new reference picture. Figure 4.11 shows how three outlines are used to develop an average suggestion. The three pictures in the column to the left is the outlines submitted by experts. The center column presents the outlines when they are stacked on top of each other in the order they have been submitted, and the right column presents the calculated average. In the first row, only the first image is applied to the algorithm, as there is no difference in any pixels, it has not changed in the upper right corner. In the second row, and the average is now different as it is an average between outline one and outline two. In the third row, the final outline is added, in the center of the row, all three together is presented, and even though they are similar, there is enough of a difference to benefit from a combination. To calculate the last average suggestion, the average suggestion from row two is used as a reference, creating the average between it and the new outline on the left in row three.

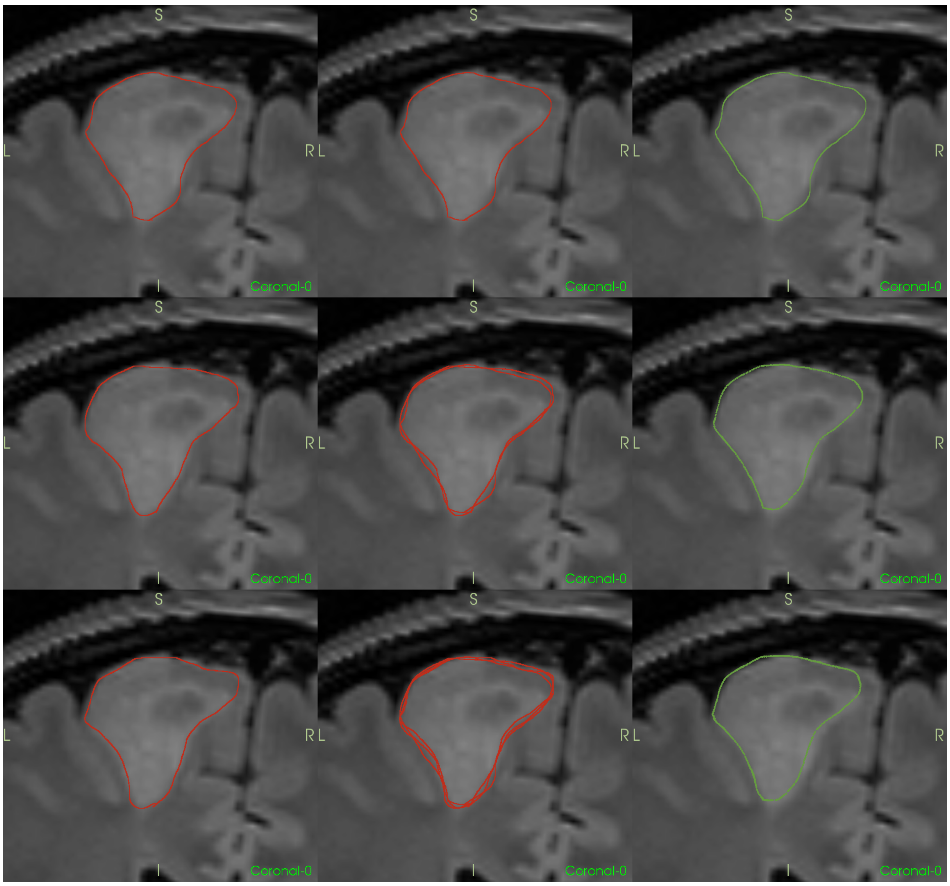


Figure 4.11: Average process

Performance

The average solution calculation is done on the server side of the application. The base implementation without any optimization searches for all the shortest distances between pixels in the outlines, which is a slow process. To reduce the amount of lookups that needs to be done in the two images, the search space was reduced by only looking for other pixels in a fixed area in the adjacent image as presented in Figure 4.12. By making the assumption that most images would be quite similar as they are submitted by experts, the search space can be reduced to a fixed length in every direction from the coordinate in the reference picture. This assumption is considered safe as images with significant differences would be considered outliers and be eliminated by experts doing a visual inspection of the images.

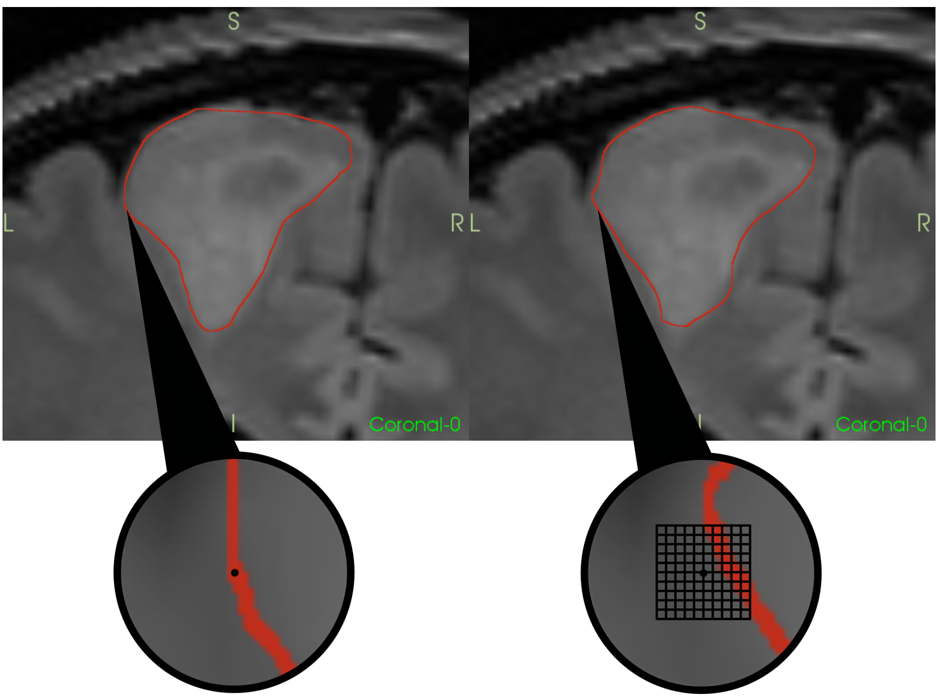


Figure 4.12:
Left: Reference image with a selected pixel direction.
Right: Image showing a search grid five pixels in each direction.

4.2 Contribution to open source software

The application is developed using multiple open source libraries and frameworks to improve the quality and improve release time of new builds. Most of these software packages are quite mature, stable and had implemented everything needed, but there were some exceptions. Using these libraries gave the opportunity to give back to a community that has provided quality solutions for the project and allowing the application developers not to re-implement or reinvent solution that has been perfected over years of hard work. In the following section contributions to the used open source software will be discussed.

4.2.1 Elm Canvas

For tasks requiring interaction with images, using a 2D canvas[1] in the web browser is required. The elm project has not yet implemented this in the standard library, but a third party library named Elm Canvas[2] is working on an implementation.

The Elm Canvas project is currently considered a working alpha release with most features of the canvas specification implemented. However, when developing the application, a critical bug was discovered as well as the lack of touch events (50) to detect interaction from smartphones and tablets.

[1]`https://www.w3.org/TR/2dcontext/`
[2]`https://github.com/Elm-Canvas/elm-canvas`

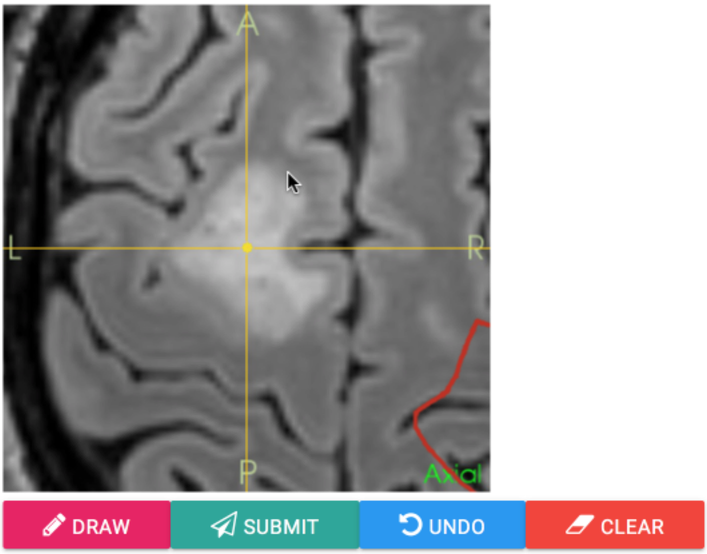


Figure 4.13: Coordinate calculation incorrectly reported

The critical bug was found in the coordinate calculation when processing a click on a given canvas. The calculation would work correctly as long as the web page with the embedded canvas was not scrolled or zoomed. If the page was scrolled or zoomed, the coordinate of the cursor was reported incorrectly with an offset equal to a number of pixels skewed by the zoom, or by the scroll as shown in Figure 4.13. This problem was resolved by taking zoom and scroll offsets into account, and a patched version of the coordinate calculation was suggested to the maintainers of the project. After confirming the existence of the bug and testing the patch, the new code was merged into the project[3].

An effort to extend the software library was made to provide support for touch events when interacting with a canvas. At the beginning of development, the canvas library only supported mouse interaction which would not work on touch devices when drawing on a canvas or interacting with the canvas in a multi-touch manner. This contribution ended in increasing the available Events API[4] with functions to cover both multiple and a single

[3] https://github.com/Elm-Canvas/elm-canvas/commit/e5873b

[4] https://github.com/Elm-Canvas/elm-canvas/blob/master/src/Canvas/Events.elm

touch of all events defined in the HTML Touch specification (50).

As a result of these contributions, the canvas now properly functions if the web page is scrolled or the page has been magnified. In addition, the canvas can now be used interactivly due to the implementation of touch events. Adding this fixes and features allows for the implementation of more advanced features, such as zooming within the canvas frame using multiple inputs on mobile devices.

Zoom on mobile devices

On mobile devices with small screens, drawing with high accuracy can be a demanding exercise. Therefore, a lot of focus was put on developing zoom functionality for these types of devices.

Efforts were made to adopt familiar gestures already used in mobile applications, recreate the user experience of a native application. A good example is the “pinch-to-zoom” gesture which is used to zoom images, web pages and maps on most phones today. “Pinch-to-zoom” is a gesture where the user puts two fingers on the device screen’s surface and moves the fingers closer together or away from each other. A user action where the fingers are moved closer to each other will perform a zoom out action while moving the finger away from each other would magnify the content.

When developing native applications for specific platforms, platform vendors like Apple and Google offers APIs for developers to quickly integrate these features. When developing a web-based application using a canvas, these APIs are not available, and only the basic touch events mentioned in the last section is available. Zoom functionality was hence implemented from scratch. A state is representing how the canvas must render the current image, containing information on how much magnification has been applied and what part of the image that is currently in the viewport.

The magnification of an image changes when the zoom mode is on and the user touches the canvas with two fingers. By performing a "pinch-to-zoom", pre-defined canvas size will remain the same while the underlying image will be scaled up. The user will then see a magnified section of the image. In Figure 4.14 an image magnified by a factor of $2.6x$ is shown. The square marked "viewport" shows the portion of the image the user can see at a given time. The viewport is the current state of the canvas and is represented as a simple data structure shown in Listing 4.2. This datastructure contains the position data needed to render the viewport correctly.

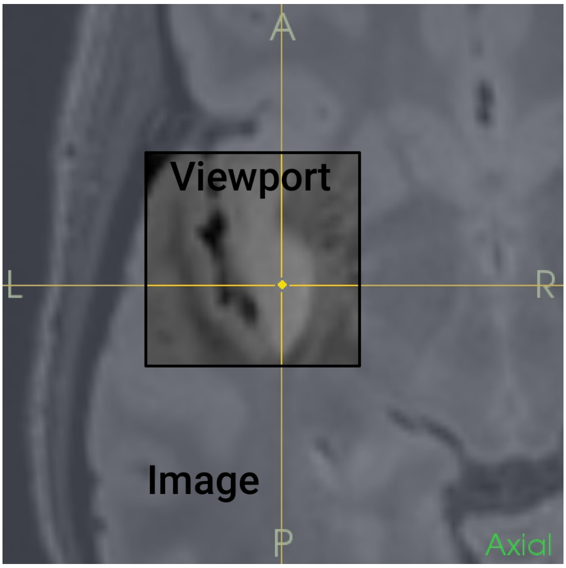


Figure 4.14: A mobile view where the canvas is zoomed in by a factor of $2.6x$ and the underlying image.

```
{ canvasSize = { width = 380, height = 380 }
, imageSize = { width = 1000, height = 1000 }
, position = { x = -256, y = -264 }
, scale = { x = 2.6315789474, y = 2.6315789474 }
}
```

Listing 4.2: Canvas zoom state

Moving the canvas would put the state through a transformation function taking into account the movement of the user's input. Based on the direction and the number of pixels traveled, the canvas position offset would change correspondingly. The applications reactive event loop register the canvas state, redrawing the canvas according to the new changes. A similar computation is calculated when zooming the image, but now the scale is a factor in addition to position.

Drawing in zoomed state

In comparison to the state holding the zoom information, drawing data is saved as a set of instructions on how the canvas must be manipulated to show the proper drawing. The drawing at any given state is stored in a list of *Drawing operations*.

Drawing operations are handed to the canvas and performed sequentially. In Listing 4.3 is a truncated example of the drawing state is presented. In the given example, the canvas will draw two very tiny lines of only a few pixels.

```
drawOperations =
    [ BeginPath
    , LineWidth 3
    , StrokeStyle (RGBA 192 47 29 1)
    , LineCap "round"
    , SetTransform 1 0 0 1 -0.8870 0
    , MoveTo (Floats 156.88 217)
    , LineTo (Floats 156.88 218)
    , LineTo (Floats 156.88 220)
    , SetTransform 1 0 0 1 0 0
    , SetTransform 0.4721 0 0 0.4721 -0.8870 0
    , MoveTo (Floats 325.89 341.89)
    , LineTo (Floats 325.89 345.89)
    , LineTo (Floats 326.89 355.89)
    , SetTransform 1 0 0 1 0 0
    , Stroke
    ]
```

Listing 4.3: Canvas draw operations

An interesting thing, however, is the modifications of the matrix using the SetTransform instruction. A SetTransform instruction will modify the matrix holding the pixel data. In the case of this application, the matrix is modified to render lines according to the current state of the zoom. When calculating how a line must be drawn, it is important to take into consideration how the image was zoomed at the point of the drawing. As this must be considered for every zoom level, the scale values is saved when the user draws on the canvas at a given zoom level. Calculating the scaling for the SetTransform operation is done by:

$$matrixScale = \frac{currentZoom}{zoomWhenDrawn}$$

Where *matrixScale* is the factor between the zoom state at the current moment (*currentZoom*) and the zoom state the application was in at the time of recording (*zoomWhenDrawn*). The calculated value is then used to modify the matrix so the following drawing operations can be drawn by using the coordinates created when the user was zoomed in. Referring back to the state in Listing 4.3 going through the drawing of one line, starting at line 6, the matrix is transformed into the right scale based on the current zoom and given an offset to start the drawing. In line 7, the MoveTo instruction tells the canvas to move the starting point of the next operation to a given coordinate, and line 8 and line 9 will draw the line from the coordinate of departure and to the end. Finishing with line 10, the matrix is reset to the original position making the canvas ready for the next rescale.

In Figure 4.15 two different zoom levels with attached drawings are presented. Each of the marked squares represents a transformation of the matrix in the canvas. To the right in the figure, the viewports from the two zoomed in scenarios are shown.

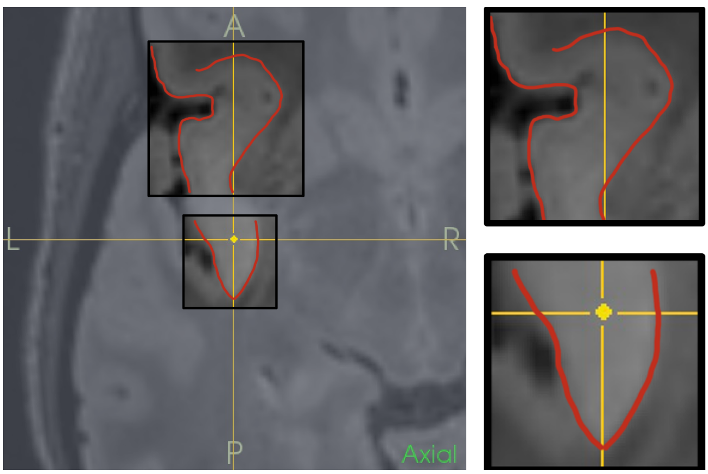


Figure 4.15: Two scaled matrices from two different zoomed viewport scenarios.

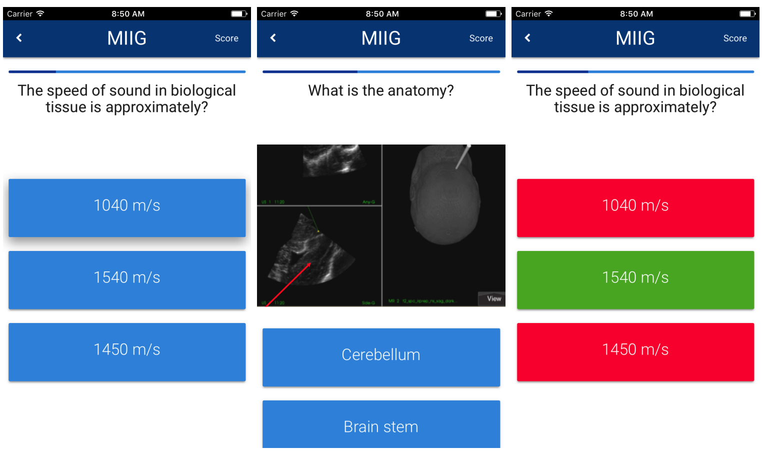
4.3 Game modes

The game has been implemented with three different game modes. In the following section, a brief explanation will be given.

4.3.1 Multiple choice questions

The game implements a multiple choice game mode, where the user answers questions to test their knowledge. The game mode is started by selecting the desired number of questions to use in the question set. The user can select as many as they want, up to the maximum available number. The questions are randomly chosen from the available data set provided by the back-end service.

A question can be either plain text, picture or video followed by three answers, see Figure 4.16. Only one of the three answers can be correct, and the presentation of the answers is randomized each time the question is shown to the user. When a user answers a question, the correct answer and the two wrong are shown in green and red (4.16c). New questions are added to the back-end service, provided as a REST API from MIIC. Adding new questions is described in detail in MIIC (16).



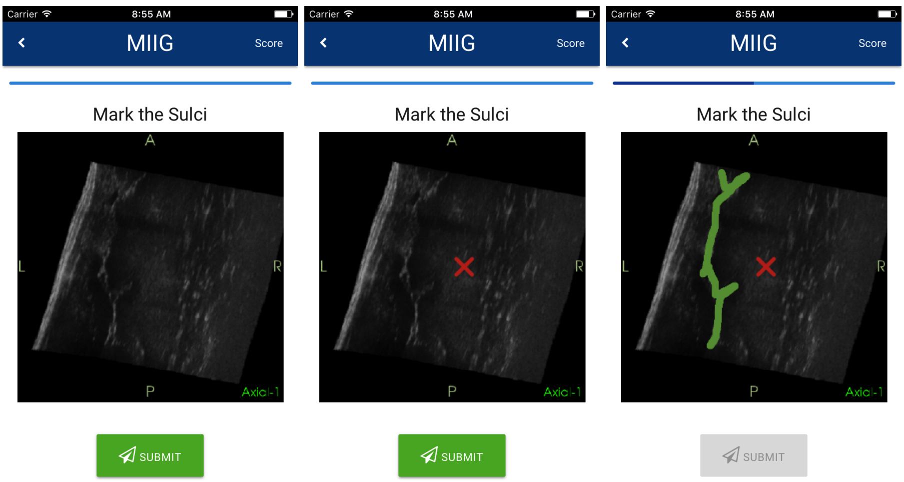
(a) Plain text based multiple choice question **(b)** Image based multiple choice question **(c)** When showing the solution

Figure 4.16: Multiple choice questions

4.3.2 Landmark questions

Landmark questions are based on selecting a point within the predefined region in an image. This game mode is considered to be an easier task than outlining the region. A landmark question is typically created from a database of images, where the target region is filled with color.

Visually, the questions are presented as a medical image as shown in Figure 4.17a. The selected point is represented by a cross as shown in Figure 4.17b and the filled region is displayed when the users' answer is submitted as shown in Figure 4.17c.



(a) Landmark question **(b)** Landmark with users' input **(c)** With solution is displayed

Figure 4.17: Landmark

4.3.3 Outline questions

In outline question mode, the player is presented with a ultrasound image and asked to draw an outline around one or more regions in the picture as shown in Figure 4.18. The regions are typically tumors, but can also be other structures. Regions are defined by a group of experts, to make sure that the tasks are of high quality, correct and taken from real world cases.

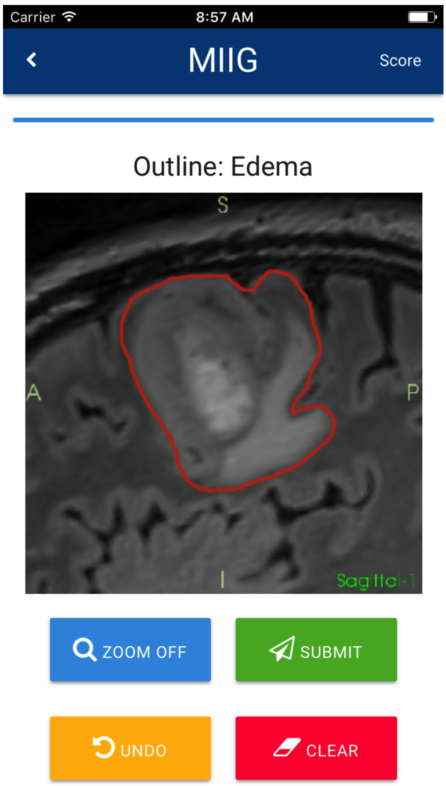


Figure 4.18:
Outline question with user input

The team of experts can create new outline questions in the MIIC software, where they can create a database of images and choose several ways to create new questions as covered in Section 4.1. A typical question contains a title, and an image.

4.4 Scoring

The application implements a scoring system based on points. A user accumulates points by completing tasks from each game mode. One task can provide a maximum of 100 points, and the calculation of these points are different for each game mode.

4.4.1 Multiple choice questions

The scoring implementations for multiple choice questions are implemented by giving 100 points for a correct answer and zero for a wrong answer. As there is no room for an answer to be almost right or a little wrong, there is no other amount of points given in this mode.

4.4.2 Landmark questions

When scoring a landmark question, the score is determined as correct or wrong, giving one hundred points or zero points. A correct answer is a mark within the area defined by an expert. A wrong answer is every point outside the predefined area.

The solution is represented as an image with the same aspect ratio as the original image but contains only data in pixels outlining the solution. A correct pixel is a pixel containing a color represented by a Red, Green, Blue and Alpha (RGBA) value in a PNG (51) image. An incorrect pixel will hold zero in all value fields of the RGBA representation. In Figure 4.19 an image with a correct answer is shown together with the underlying solution to the task. Both frames are the same size which opens up for a direct lookup on the same coordinates in the solution frame. Direct lookup in the two-dimensional array allows checking a submission without requiring significant computational power, allowing good performance on low powered mobile devices.

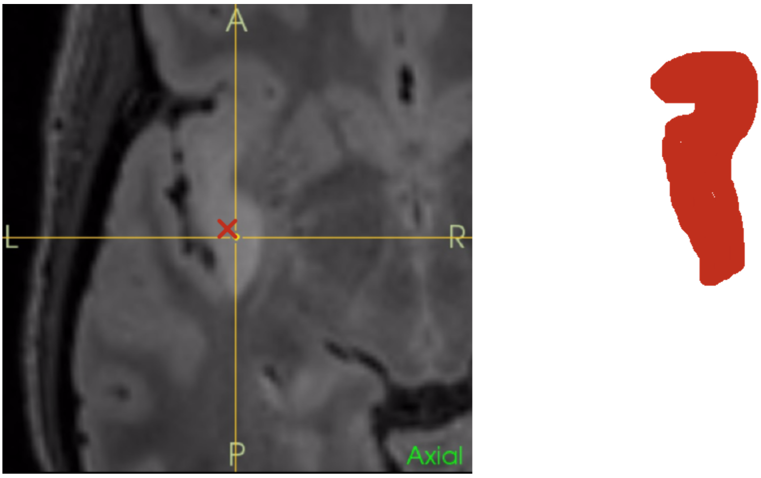


Figure 4.19:
Left: a correct answer by a user
Right: the representation of a correct solution

4.4.3 Outline questions

Scoring outline questions is implemented by calculating the average Euclidean distance[5] between two outlines. The first outline is the reference outline, the solution, which is provided by the backend service. The second is the outline drawn by the user as their suggestion. The score between 0 and 100 is then calculated by a linear scale. The implemented Euclidean distance algorithm takes the coordinates from the user submitted outline and calculates the distance to the closest pixel in the solution outline. The shortest distance calculated from each pixel is then accumulated together before it is divided by the amount of pixels in the solution. This is done to create the average distance from the user created outline to the solution outline.

The score is then calculated from the average Euclidean distance based on a linear scale presented in Figure 4.20. The Euclidean average is represented as a number of pixels that differ from the original outline, where zero indicates exactly the same shape. The outlines has to be very similar to get a high score because the linear scale is quite strict, giving zero points if the outlines differs more than 10 pixels on average.

[5] https://en.wikipedia.org/wiki/Euclidean_distance

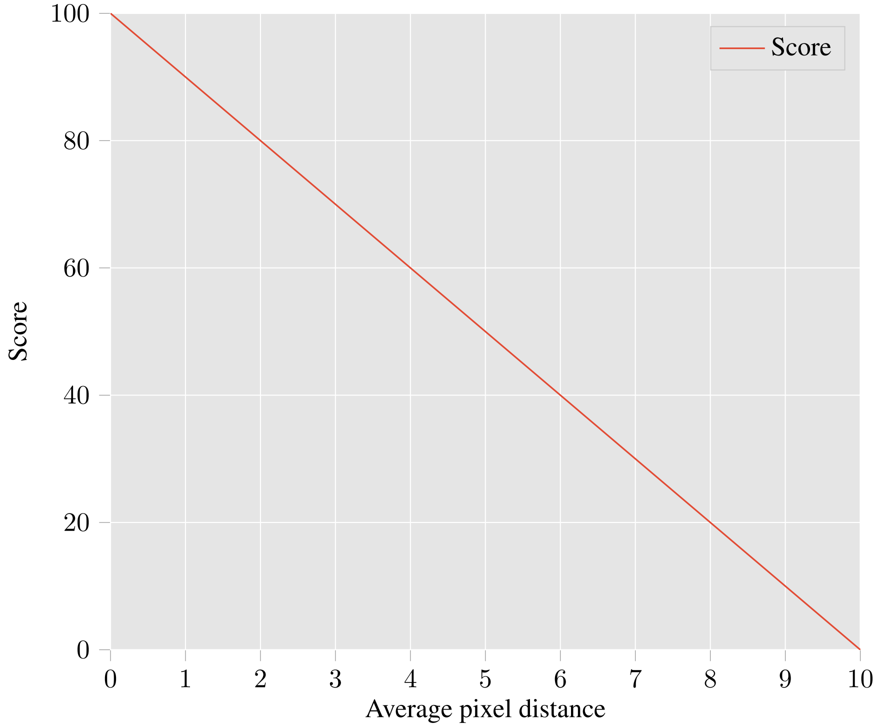


Figure 4.20: Linear cutoff for Outline scoring

When the pixels is retrieved from the canvas, the image representation is taken into consideration to prevent a skewed result from different platforms with different screen sizes and resolutions. For example, looking at an iPhone mobile device and a desktop computer, the size of the rendered canvas would be significantly different as shown in Figure 4.21. There are two problems to take into consideration in this scenario. Retrieving a drastically bigger picture will make the computation of the Euclidean distance a lot slower as the computational complexity of this algorithm is $O(n^2)$. In addition, the average pixel distance would be inconsistent across platforms and screen resolutions as the actual picture will have a higher or lower pixel count. In the following sections these two issues will be explored.

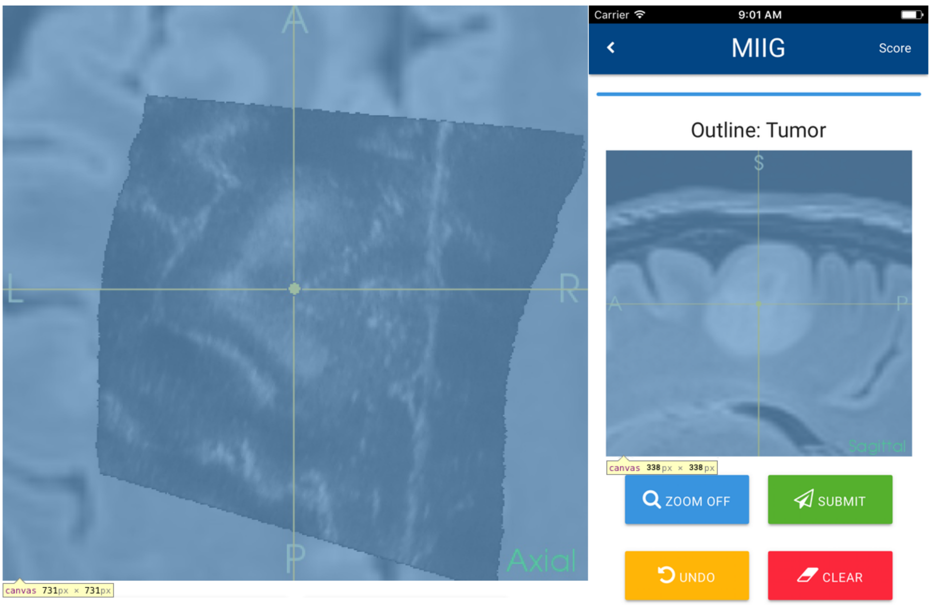


Figure 4.21:
Left: A 731×731 rendered canvas on a desktop computer
Right: A 338×338 rendered canvas on a iPhone 6

Performance

As scoring calculation is executed on the user's device, optimizing the performance to execute swiftly on low power mobile devices is a crucial part of the implementation. Therefore different approaches to optimizing the execution time were researched. The most efficient way to make a quadratic time algorithm faster is to reduce the amount of data that needs to be considered in each iteration. In the case of this program, the amount of pixels in each iteration may differ significantly from platform to platform depending on the resolution the platform uses and how large the canvas is rendered.

The approach chosen to achieve lower execution time was to reduce the quantity of pixels in the reference and the user submitted outline. The idea of reducing the input data is not new and is also used in both MIIG and MIIC. However, there are some problems with the previous implementations when we consider mobile devices and reliability. In the next sections, an overview of the scoring in MIIG and MIIC will be described, followed by the changes made in MIIP. From there, a comparison on how the different scoring algorithms work on various screen sizes will be presented with an example.

In the MIIG system, the scoring algorithm will look at all pixels in each list. When the system checks two pixels against each other, and if the distance is smaller than 10 pixels, it is automatically considered the minimum. Meaning all pixels after that point, even if they are closer to each other than 10 pixels, will not be considered. Using this technique effectively speeds up the algorithms by allowing significantly fewer iterations in most cases.

MIIC uses a different approach. To reduce the number of iterations only every 20 pixels in the inner loop is considered, effectively reducing the inner loop by $1/20$.

MIIP's scoring implementation does not remove pixels in the actual calculation. To prevent inconsistency between different screen sizes, all pixels in both outlines are considered, giving the definite Euclidean distance average between the outlines. As this does not reduce the amount of pixels by itself, it is quite slow and further optimization will be discussed later.

To measure the various scoring algorithms against each other, a set of tests were developed. The tests are constructed as a set and executes each algorithm on a canvas with a user submitted answer and the solution at a given resolution ranging between mobile and desktop. Each test consists of one image with an answer submitted by a user and an expert solution. The image is scaled to a range of different resolutions, ranging from 200 pixels by 200 pixels to 1000 pixels by 1000 pixels to represent all the different platform resolutions we can encounter. For each resolution, each algorithm is used to calculate a Euclidean average for the two outlines in the image. The test was executed on four separate images with four different outlines. In the next section, one case will be presented, and the next three is available as attachments A.

Figure 4.22 shows a user submitted answer with a big diversion from the solution outline. The first thing to notice is that when the image is rendered on a desktop, the size of the canvas is larger than on a mobile device. As a result, the number of pixels is lower on the mobile device. To be exact, when reviewing the number of red pixels in the user submitted outline, the amount from the desktop is 1829, almost twice the amount of red pixels from the mobile device, which is 775.

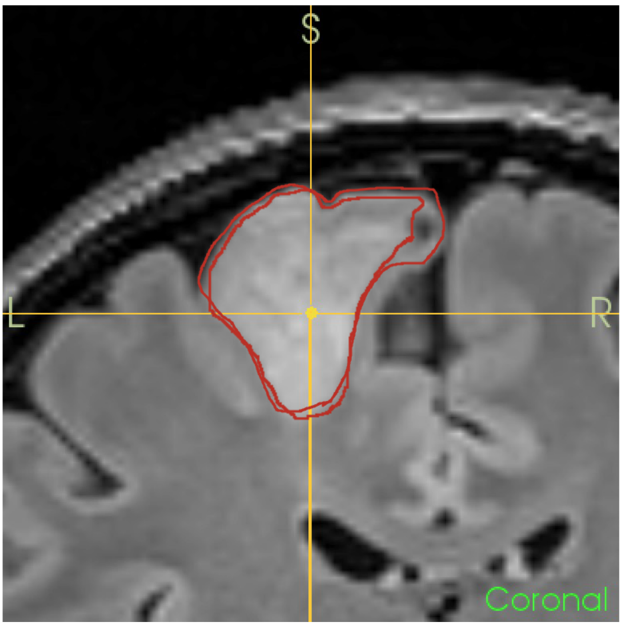


Figure 4.22:
Given a rendered canvas at 733×733, the amount of submitted pixels are 1829 while on a 338×338 resolution, the amount is 775

Not only will this drastically change the execution time of the scoring calculation, especially since the solution outline has fewer pixels in its outline. Considering a smaller image will also yield a different Euclidean distance, were the average distance between two images is smaller in a low resolution image than in a high resolution image. As shown in Figure 4.23, the pixel difference between to coordinates is scaled alongside the image.

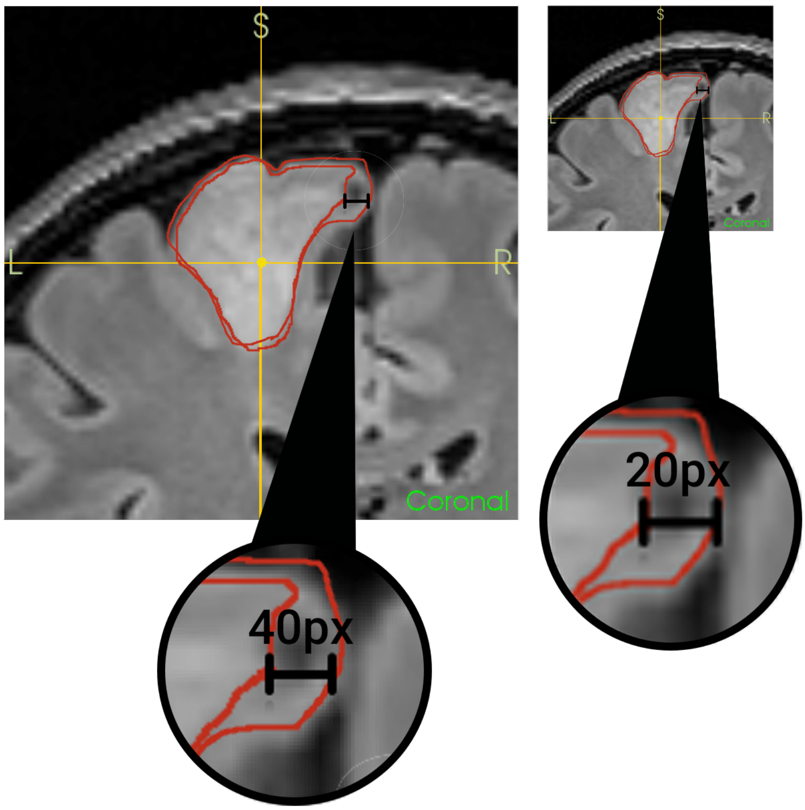


Figure 4.23:
Left: A canvas rendered on the desktop yielding 40 pixels
Right: A canvas rendered on an iPhone 6 yielding 20 pixels distance distance

Taking into account how the Euclidean distance must be smaller at the lower resolutions, the test between the three scoring implementations yields an interesting result. In Figure 4.24 and Table 4.1 the results from executing the different algorithm on a series of resolutions is presented. The first thing to notice is how MIIG's algorithm does not scale well for smaller screens, where it is heavily bound by the fact that 10 is automatically set as a minimum value. Given the noticeable difference between a resolution of 1000 by 1000 and 200 by 200, the Euclidean distances should have been significantly lower on the smaller screen.

When considering MIIC and MIIP, the results are closer to the expected value. The Euclidian distance is reduced in a linear fashion making it the same accuracy on each resolution. MIIC, however, has in the given test case one outlier when the resolution is 600 by 600 pixels. Based on the implementation, skipping pixels at a fixed interval will in some cases remove necessary data to provide an accurate average. When considering canvases of larger sizes, it can be argued that the implementation will not lose enough data to make any difference. However, a fixed interval of skipping every 20th pixel can prove to be quite significant on smaller resolutions.

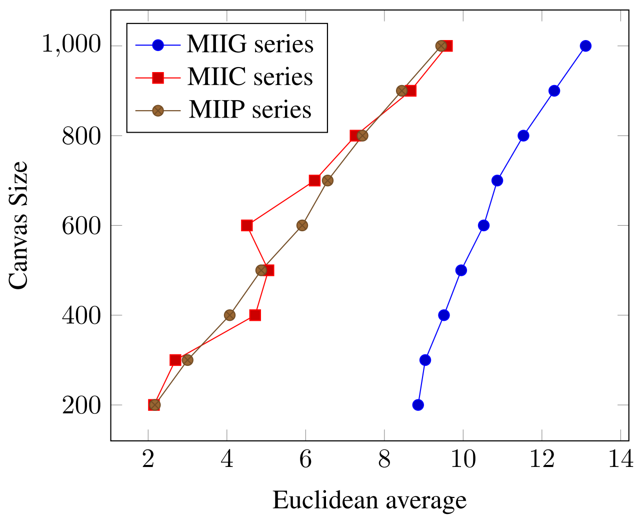


Figure 4.24: Plotted result from each Euclidean average algorithm on different resolution

Size	**MIIG**	**MIIC**	**MIIP**
1000	13.11176602827905	9.587571330201351	9.440917731967899
900	12.314875108731204	8.66689451108579	8.442206466890859
800	11.53134667987292	7.267934954694367	7.444504761101635
700	10.86691078962241	6.229772886510186	6.560029412895148
600	10.523191825042671	4.508216627726428	5.913251676763076
500	9.949874166807351	5.054268223118228	4.871410197932933
400	9.512791575045698	4.717082688542871	4.072299981952993
300	9.037602692336344	2.6929527312693473	3.0018172606537608
200	8.856359296407705	2.1475324561732762	2.1722259191732762

Table 4.1: Results from each Euclidean average algorithm on different resolution

Consistency

Analyzing the results from testing the different algorithms shows that each scoring function is quite stable on the canvas sizes rendered on desktop computers. Which is a fine result for both MIIG and MIIC, as both mainly focuses on this platform. Nevertheless, when considering MIIP, which is developed to be cross-platform and focuses on mobile usage, the inconsistencies across canvas sizes is not acceptable. As stated earlier MIIP does consider all the points in both outlines, which solves this problem. The next challenge is considering all points, rapidly, and in a general way on all platforms.

A measure taken to provide a more swift calculation of the Euclidean average is to scale down the image to a lower resolution before executing the average calculation. To prevent loss of critical data, both the solution outline and the outline submitted by the user is scaled down by using a transformation matrix to reduce the size of the image. By reducing the image, the amount of points that needs to be considered in the calculation is reduced evenly across the frame without losing information about how the outline is shaped as shown in Figure 4.25.

Looking at Figure 4.25, the bottom outlines represents the canvases created in the background. These canvases are rendered with a size of 200 by 200 instead of the original 733 by 733. The bottom outlines is scaled up to accentuate that the lines in the shape are not as smooth as in the original. However the shape of the outlines is still as accurate as the original outline. By rendering the calculation canvases in the same size on each platform, independent of their original size, we also eliminate the problem where a Euclidean distance is greater on a larger screen as discussed in Section 4.4.3. Getting a similar range of distances on every platform allows the use of a linear scale for calculating the score range between 0 and 100 described in Section 4.4.3.

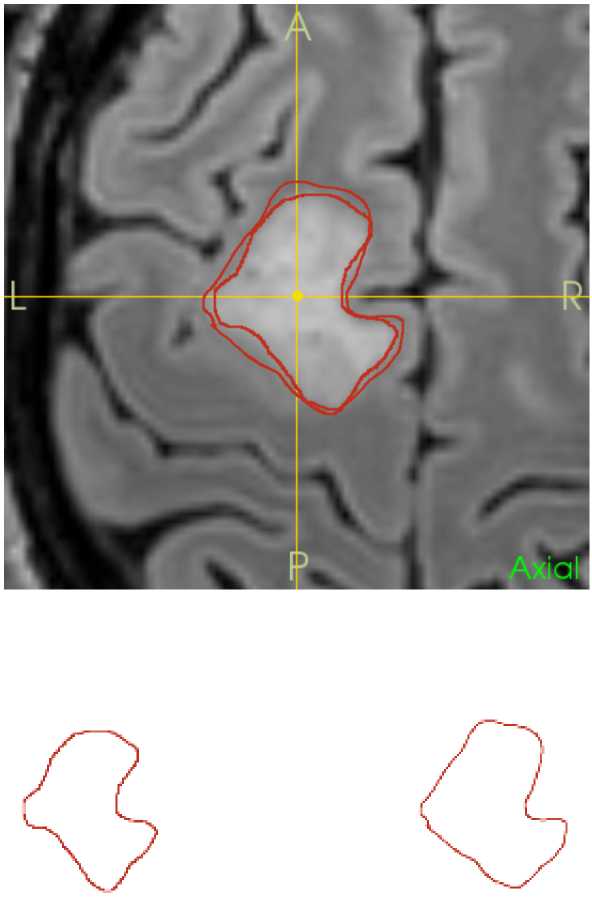


Figure 4.25:
Top: A desktop canvas viewed by a user
Bottom: Calculation canvases rendered in the background, scaled up from 200 by 200

Chapter 5

Results

In this chapter the results from the project will be presented. The first section will focus on the result of the development, the game and administration interface. The second section will focus on user feedback and the result from user testing of the game.

5.1 The game

The research and development resulted in a web application with an emphasis on easy installation to user's mobile phones. The application has a more native feel than a web page and is well integrated with the different mobile operating systems available today. The application also works well on desktops as well as tablets using the same browsers and mobile operating systems.

5.1.1 Mobile devices

The application has been tested on a wide array of modern smartphones as shown in Table 5.1. These devices cover the most common phones used in the market today with the adequate screen size for this type of application. In general, even though not tested, the web technology used should work on phones with multiple versions lower than the lowest presented in the table, with the exception of the one phone with Windows Phone.

Phone	**Operating system**	**Browser**
HTC One M8	Android 6.0	Google Chrome
HTC One M7	Android 5.0.2	Google Chrome
Google Nexus 5X	Android 7.1.2	Google Chrome
Samsung Galaxy S4	Android 5.1.1	Google Chrome
Nokia Lumia 810	Windows Phone 8.1	Microsoft Internet Explorer
Apple iPhone 7 Plus	iOS 10.2	Apple Safari
Apple iPhone 6S	iOS 10.2	Apple Safari
Apple iPhone 6	iOS 10.2	Apple Safari
Apple iPhone 6	iOS 10.2	Google Chrome
Apple iPhone 5S	iOS 9	Apple Safari
Apple iPad	iOS 9	Apple Safari

Table 5.1: Tested mobile devices

5.1.2 Browser support

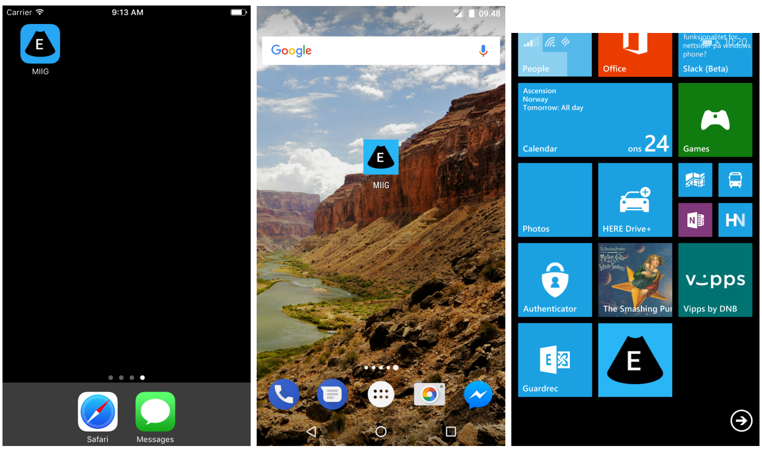
As the application must work well on desktop as well as mobile, testing the compatibility with the most commonly used browser was tested on the most common operating systems. In general, the application can run without problems in every modern browser on every major operating system. The only browser found to not work properly was *Microsoft Internet Explorer* from version 9 and lower (Table 5.2). As this browser is mainly deprecated in favor of newer versions, this is not considered a major concern.

Browser	**Operating System**	**Supported**
Google Chrome	Windows, Linux, macOS	Yes
Microsoft Edge	Windows	Yes
Microsoft Internet Explorer 10 & 11	Windows	Yes
Microsoft Internet Explorer 9 or lower	Windows	No
Apple Safari	macOS	Yes
Mozilla Firefox	Windows, Linux, macOS	Yes
Opera	Windows, Linux, macOS	Yes

Table 5.2: Supported desktop browsers

5.1.3 Integration

To give the user a native feel of the application, an approach to make the application "installable" to the users' home screen was taken. Figure 5.1 shows how the icon for the application integrates with the home screen on each platform. In Figure 5.1a the Apple iOS home screen is shown with the icon in the upper right corner. In Figure 5.1b the Android home screen is shown with the icon in the center. And in Figure 5.1c the "tile" in the Windows Phone start menu is located in the center at the bottom. To open the application, the user simply has to touch the icon.

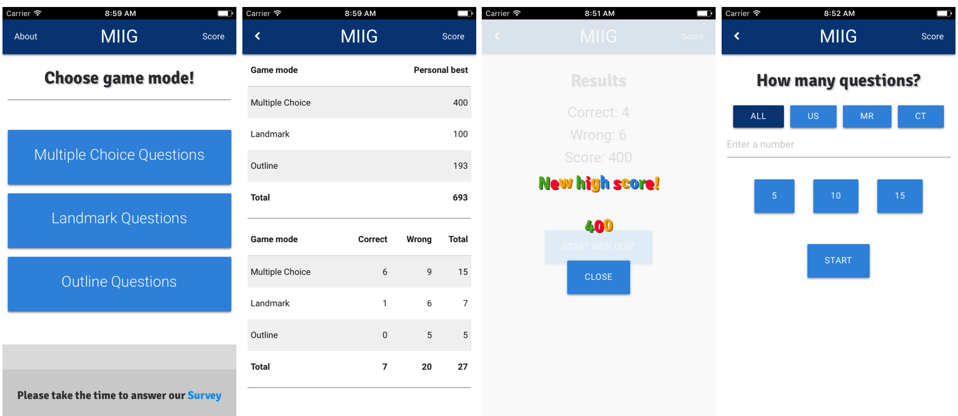


(a) iOS **(b)** Android **(c)** Windows Phone

Figure 5.1: Application integration

5.1.4 Generic views

The application contains a set of four generic views used across game modes and at the core of the application. When the application is opened, the user is presented with the main view, or the startscreen (5.2a). This screen is the entry point to all other parts of the system. From here, the three different game modes and the scoreboard can be accessed. The scoreboard can be accessed from the upper right corner, and this view is shown in Figure 5.2b. This score is accumulated over time. For each game mode, whenever a user hits a score that is better than their previous high score, the new high score is displayed (5.2c). The last generic view in the application is the view for selecting a number of questions (5.2d). For the Landmark and Outline game mode, this view also gives the opportunity to select a modality. Users only wanting to practice Ultrasound, MR or CT can choose a specific mode, or they can select all. This view can be considered the start point for each mode, and it will also present the user with a suggested amount of exercises based on the available dataset.



(a) Startscreen | **(b)** Scoreboard | **(c)** Highscore | **(d)** Questions

Figure 5.2: Game mode views

To provide users with adequate feedback, error messages will be displayed if the application fails. Such messages will be returned e.g. if the backend server is unavailable or if a user requests more questions than available, as shown in Figure 5.3. Giving proper information is important to prevent users from losing confidence in the application and to intuitively understand what is happening, and to avoid misunderstandings.

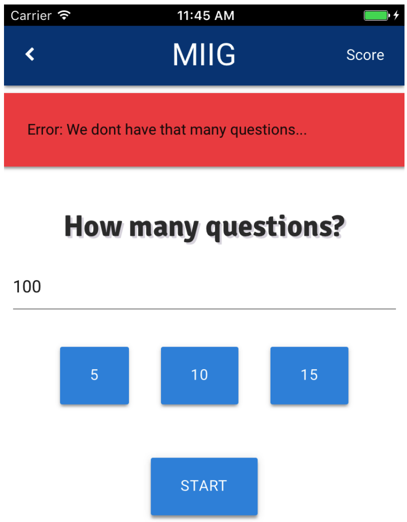


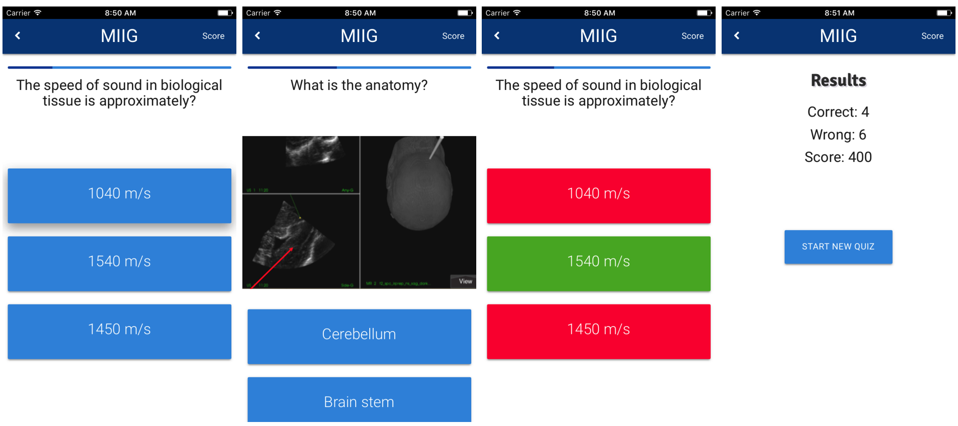
Figure 5.3: Error message

5.1.5 Game modes

In the following sections, the game modes of the application will be presented.

Multiple Choice

The first game mode in the application is the Multiple Choice game mode. After the game has been started, the first view will present the user with either a multiple choice question (5.4a) or a multiple choice question with an image (5.4b). This view contains a question for the user to answer, three possible answers and a progress bar to show the user how far they are in the exercise set. After an answer has been selected, the solution view (5.4c) will be rendered for the user for three seconds. When the user has finished the requested set of exercises, the result view (5.4d) is shown to the user.

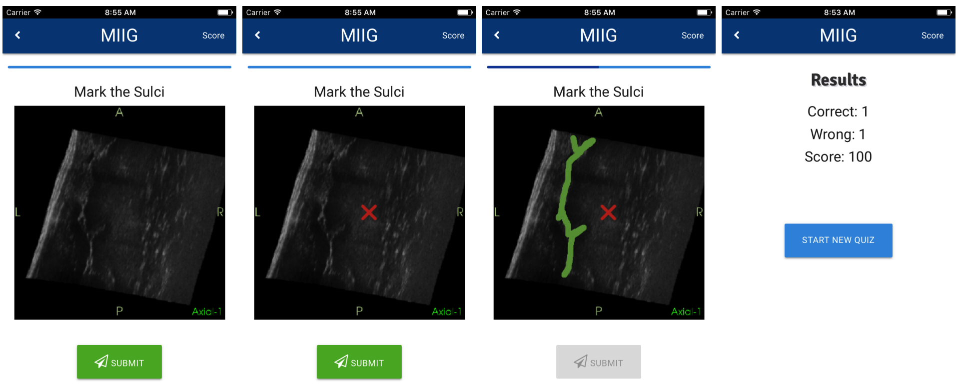


(a) Multiple Choice **(b)** With image **(c)** With solution **(d)** Results

Figure 5.4: Multiple choice views

Landmark

The second game mode in the application is the Landmark game mode. Entering this game mode will render a canvas with a medical image (5.5a) for the user to point out the structure given by the exercise text, for example the sulci. When the user have clicked on the canvas, the spot is marked with an "X" (5.5b). The user can then submit the answer by clicking the submit button and the view changes to show the solution (5.5c) for three seconds. When the complete set of exercises has been marked, the results view is rendered as shown in Figure 5.5d.



(a) Landmark

(b) With user input

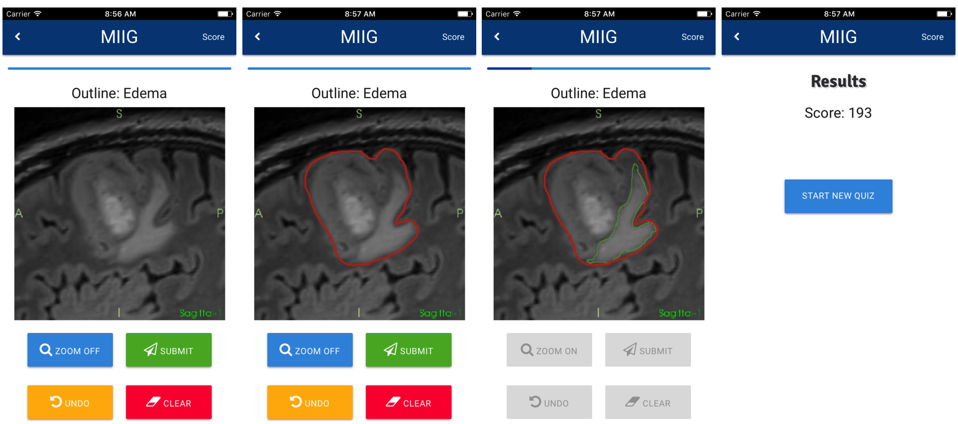
(c) With solution

(d) With results

Figure 5.5: Landmark views

Outline

The third game mode in the application is the Outline game mode. Selecting this game mode will take the user to a similar view as a landmark question, with the addition of three new buttons as shown in Figure 5.6a. This view allows the user to draw an outline on the canvas and the user can zoom to magnify the image for better precision. The initial mode of the canvas is drawing mode, which means that the user can draw, but not zoom. The user is presented with multiple options as buttons below the canvas. The buttons offer the following functionality; the blue button "Zoom Off" will toggle the zoom mode. If the zoom is enabled, this button will be named "Zoom On". The green "Submit" button will submit the answer. The yellow "Undo" button will remove the last line drawn, meaning the line between a "touch down" and "touch up" by the user. And at last, the red "Clear" button will remove everything submitted by the user from the canvas. Whenever a user input is drawn on the canvas, the input is represented by a red line, and when the user submits the answer the solution is presented together with the users input, as shown in Figure 5.6c. When the set of selected exercises is done, the result view is rendered (5.6d).



(a) Outline

(b) With user input

(c) With solution

(d) With results

Figure 5.6: Outline views

5.1.6 User view flow

The following diagram (5.7) describes the flow a player would take through the system. Each view in the application can be considered a state in the core of the application. The user always starts at the “Startscreen” view.

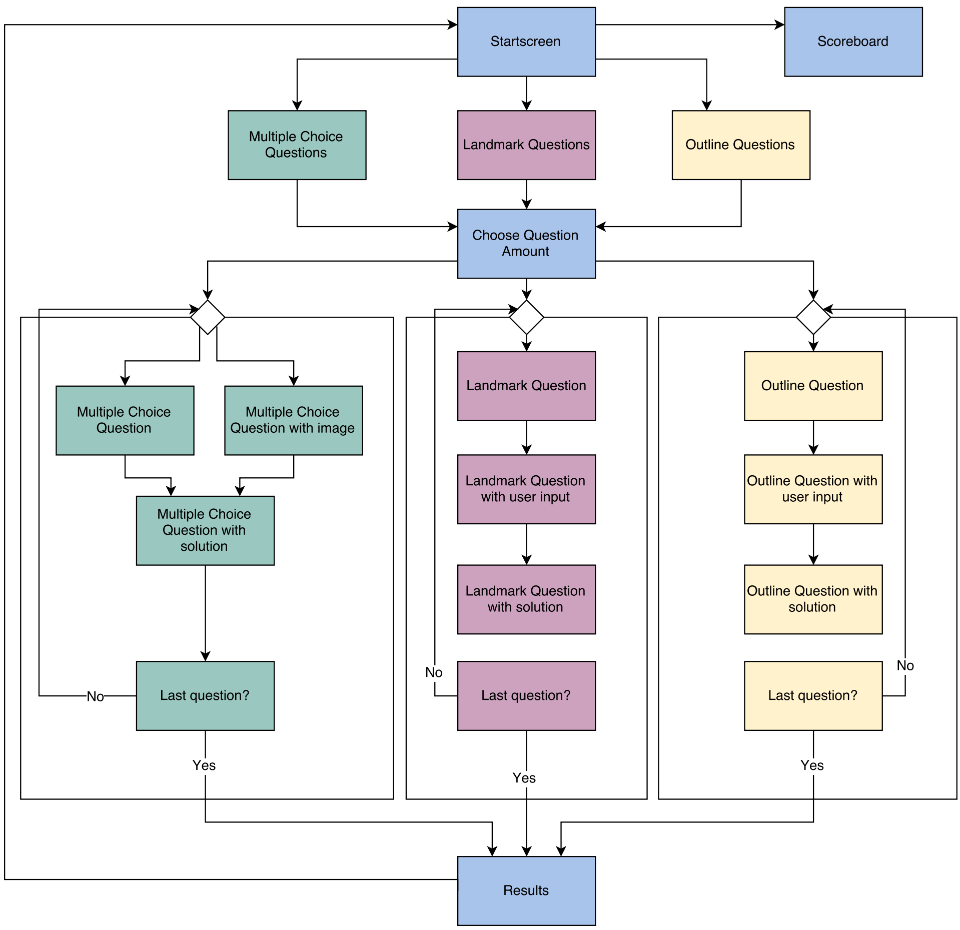


Figure 5.7:
Diagram showing the flow of the system state

5.2 Evaluation

In this section the feedback received from the user test will be presented. The user test was conducted through a online questionnaire after the users performed the install process and completed a set of tasks. The questionnaire contained two parts, one System Usability Scale (SUS) (52) for evaluating the user interface and one part targeting the usefulness of the application.

In addition to the executed user test, a large scale test will be conducted at the 9th INTERNATIONAL TRAINING COURSE: ULTRASOUND IN NEUROSURGERY conference[1] at St. Olav's Hospital in Trondheim the 15th of June 2017. The course is intended for the neurosurgeons that would like to gain more knowledge in the practical use of ultrasound in neurosurgery. These users are considered to be an important user group for this application. The results from these test would be a great supplement to the user test already conducted. However, this results will not be available before the deadline of this thesis.

[1] http://www.sintef.no/projectweb/usigt-en/courses/ultrasound-in-neurosurgery/

5.2.1 Participants

To evaluate the application, answers from several groups of different disciplines were gathered. The majority of participants were students training to be medical doctors or technologists. In Table 5.3 the split between the different disciplines is shown. As one can see, there where a few representatives from elsewhere in the healthcare industry, and some unrelated both to technology and healthcare. In the first part the results from the SUS will be presented where all participants will be considered. Followed by the medical students answer to the usefulness questions.

Discipline	**Participants**
Medical doctor	0
Medical doctor (student)	20
Other healthcare professional	5
Other healthcare professional (student)	2
Technologist	0
Technologist (student)	15
Other profession	2
Other profession (student)	1

Table 5.3: Participants by Discipline

Dividing the answers into two groups is done to get the most accurate information regarding the perceived usefulness of the application. The medical doctor students has the ability to evaluate the learning potential in the application, while all the test users can evaluate the user interface.

As stated by Jakob Nielsen, the amount of users for a quantitative user interface test should be at least twenty users to provide a reliable study (53). The amount of medical students is 20, and therefore, considering only the target audience should provide a sufficient data set.

5.2.2 Executed test

All users that responded to the questionnaire were asked to use the application by doing a set of simple exercises (Attachment C) which would cover the main functionality of the application. In addition, they where free to use the application as they wished afterward. Each user was handed a document with information about the study, an installation guide and exercises. The questionnaire can be separated into two sections. Ten of the questions belong to a standardized System Usability Scale (SUS) (52) test, which is a standardized test to provide a score of the usability of the interface tested. It was developed in 1986 by John Brooke and has then stood the test of time and proven to be very efficient, cheap on resources and quite accurate. The rest of the test is questions related to perceived learning and usefulness of the application.

Each user in the test used their own device. By using their own device, the user was able to use a system they were familiar with, not introducing more foreign elements than necessary. Each user tested the application on their own time, with no supervision. By performing the test in this way the users were able to finish the exercises in their own time without pressure from onlookers or supervisors being in the same room.

However, this model does not take into account situations where the users encounter a technical issue. Technical issues, meaning problems with their devices and not the application, e.g. trying to install the application on an unsupported device. Issues such as these can influence the overall score of the application and should be taken into account when considering the results.

Another fact that should be considered is that none of the users got an introduction or got to see the application before using it. Not getting any introductions, and the fact that the users were not able to ask for assistance with technical problems, may have affected the user experience score.

5.2.3 Results section for part one of the questionnaire

The first part of the questionnaire contained the SUS questions. These ten questions focuses on the user interface and the questions are designed to prevent users from answering without thinking, by making them consider a series of statements that covers most of the application.

In Table 5.4 the total score for each individual discipline is presented, as well as the total score for the entire system.

Discipline	**SUS Score**
Medical doctor (student)	69,38
Other healthcare professional	61,50
Other healthcare professional (student)	86,25
Other profession	72,50
Other profession (student)	60,00
Technologist (student)	75,50
Total Score	**72,84**

Table 5.4: SUS score breakdown between disciplines

5.2.4 Results section for part two of the questionnaire

The second part of the questionnaire is a series of questions regarding the functionality and usefulness of the application. As medical doctor students are considered the main users for the application their answers are shown in Table 5.5.

Question	**Average**	**Median**
I have self interest in learning medical imaging interpretation	4,30	4
I am familiar with interpretation of Ultrasound images	3,15	3
I am familiar with interpretation of MR images	3,40	4
The game is fun to play	3,05	3
Getting a high score motivated me to play again	3,20	3
The zoom functionality help me to be more accurate	2,85	3
Regular use of this application will improve my medical imaging interpretation skill	3,50	4
I felt that a challenge for my skill level was available	3,25	3
It would be beneficial for my current medical imaging interpretation skill to include this application in my training	3,30	3
I would recommend this app to a friend	2,95	3

Table 5.5: Average and median from usefulness questions

5.2.5 Textual feedback from users

In the questionnaire, a free text area was added at the end for users who would like to give additional feedback after testing the application. There was a lot of constructive feedback, giving an insight to how the user was thinking while testing the application. Most of the feedback was tied to the exercises of the application, but there was also feedback on the whole idea and the application.

Exercises

The medical students that were giving textual feedback gave a lot of opinions on the exercises. The general feedback is that the lack of context provided with each task made the exercises too hard. A common denominator was that the students would like to get the task presented as a case, with more pictures and information about what they are seeing. A few users mentioned that they were unsure about the image and organs orientation in the exercise as well. One user suggested that the application could provide an explanation to each solution. The user suggested that this would improve the learning benefits. There was also mentioned that the amount of available exercises would need to increase. To eliminate memorization of the exercises the amount of exercises needs to be large and varied.

Implementation

During testing, some user gave us feedback on the application and its functionality. In the following paragraphs, some of the feedback will be presented.

Some users reported that the accuracy of the drawing functionality was quite impressive and combined with the zoom functionality enabled the users to be very precise in their drawings. There were, however, also reports from users that could not get the zoom functionality to work. Based on the responses given, it is hard to say if this is a user error because of poor interface design by the developers or due to an unsupported browser or device. As one of the users provided us with the vendor of the phone, testing for the bug in a device laboratory may benefit further development.

A few users were also reporting that the application lacked an “Undo” and “Clear” feature. As this is implemented and available as buttons, this was considered peculiar. After investigating the issue, a reasonable assumption is that the user did not install the application on the home screen to get the “app-view” of the application. By not doing this, the phone browser may push the website so far down that these two buttons, which are rendered at the bottom, will not be visible.

Most users reported that they liked the high score system and requested even more scoring feedback during play. It was also requested more details on how points are awarded on outline questions, so they could figure out how they were penalized for a weak drawing. Summarized; even though there is potential for improvement, most users endorsed the application and the idea.

Chapter 6

Discussion

In this section, the results found in Chapter 5 will be discussed in relation to the research questions.

6.1 RQ1: How can we incentivize the use of this system?

The approach to incentivize system usage in the implemented artifact was gamification. In this section, the two statements from the questionnaire that focused on this approach will be discussed.

6.1.1 Is the game fun to play?

After the users had tested the application, they were asked to rate the following statement, *The game is fun to play*. The average result of this answer was 3.05, as presented in Table 5.5. This indicates that the users on average is fairly happy with the fun factor of the game as shown in Figure 6.1. Some users indicated in the textual feedback that they would like more game elements as they found them motivating.

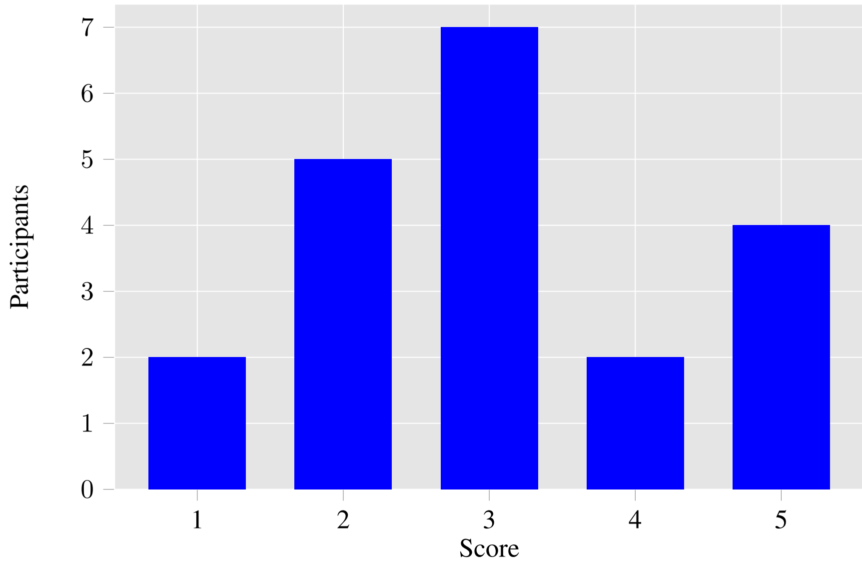


Figure 6.1:
Questionnaire statement:
The game is fun to play

6.1.2 Does scoring motivate the player?

The implementation of scoring, and more specifically the high score, was one of the core gamification elements of the game. The scoring functionality was intended to encourage users to beat their own high score, as well as comparing their high scores with friends. The users were asked to rate the statement *Getting a high score motivated me to play again*, and most users agreed with this statement as shown in Table 5.5. The average answer for all users was 3.20 indicating that this gamification element could encourage the users to play again and a score of 3.20 indicates that players were motivated by the high score.

6.2 RQ2: To what extent is the application perceived as a useful learning tool?

To make the application worthwhile, it would need to have a future as a learning tool. A serious game is not only created for entertainment proposes, but need a secondary element

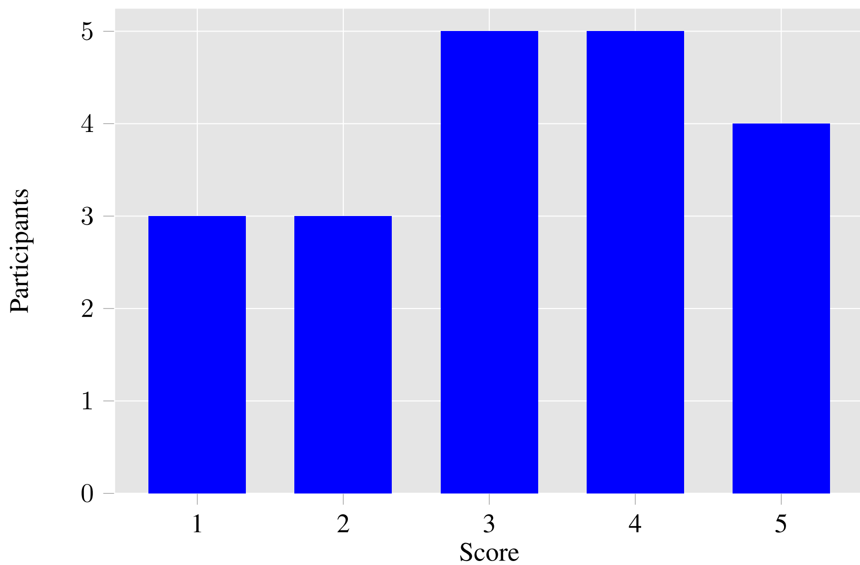


Figure 6.2:
Questionnaire statement:
Getting a high score motivated me to play again

such as an educational part. As part of the questionnaire, a great deal of focus was put on perceived learning. The test group was significantly larger and more relevant than previous attempts. And since it was important for the development of the application, that it was indeed a learning tool, they were asked to answer questions about the perceived learning and usefulness of the application.

6.2.1 Is the game perceived as a useful learning tool?

To measure the users perceived benefit of the application as a tool for learning, the test users were asked to rate two statements. The first statement, *Regular use of this application will improve my medical imaging interpretation skill*, focuses on repetition, stating that if the user would use the application on a regular basis, their current skill level would improve over time (54). The idea behind this statement is that repetition of tasks is shown to increase the skill in the task that is repeated, meaning, if you do a task over and over again you will get good at that task (54).

As shown in Figure 6.3, most users believe that the game would be beneficial for regularly exercising their medical imaging interpretation skills. The average score on this question was 3.50, suggesting that the perceived learning value is above average.

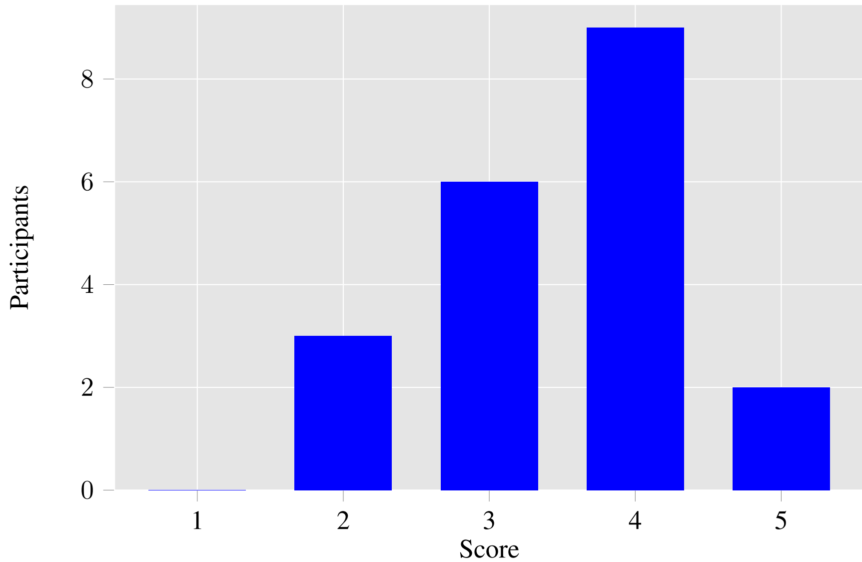


Figure 6.3:
Questionnaire statement:
Regular use of this application will improve my medical imaging interpretation skill

The other statement, *It would be beneficial for my current medical imaging interpretation skill to include this application in my training*, focuses on perceived learning and the applications ability to supplement already existing training as an additional tool. The average score for this question is 3.30, indicating that most users believe that including the application would be beneficial. As shown in Figure 6.4, there is a good amount of users suggesting that the application may improve their skill as part of their current training. In addition, the free text area on the questionnaire had a few encouraging statements indicating that this type of application was something of interest.

> *A great over all idea and something we could really need*
>
> — Anonymous Medical doctor (student)

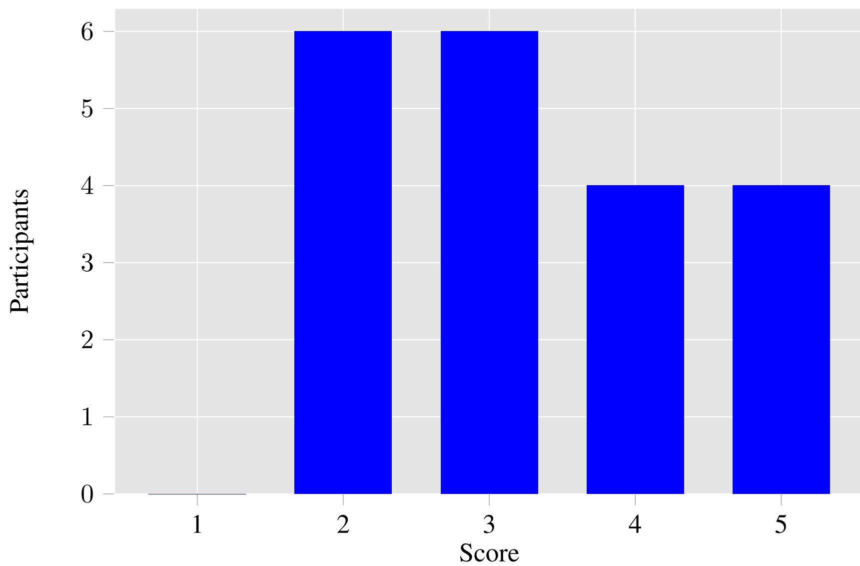


Figure 6.4:
Questionnaire statement:
It would be beneficial for my current medical imaging interpretation skill to include this application in my training

6.2.2 Does the application provide adequate challenge?

To provide a feeling of progress when learning, in addition to scoring, the application needs to provide the right level of challenge. As mention in chapter 2, correct level of challenge is important to motivate the player to continue to play. Because of this the application has three game modes. To understand to what degree the test users felt they were challenged at their skill level, they were asked to rate the following statement *I felt that a challenge for my skill level was available*. The average for this statement is 3,25 as shown in Figure 6.5. This indicates that the participants are overall positive to the level of challenge, but the results also shows that some felt they did not get an adequate challenge.

From the textual responses, as mentioned in Section 5.2.5, some people felt that they needed more information and some suggested that the tasks were too difficult for their level in general. Considering that the participants are at quite different stages in their medical degree, this can explain the spread in perceived challenge.

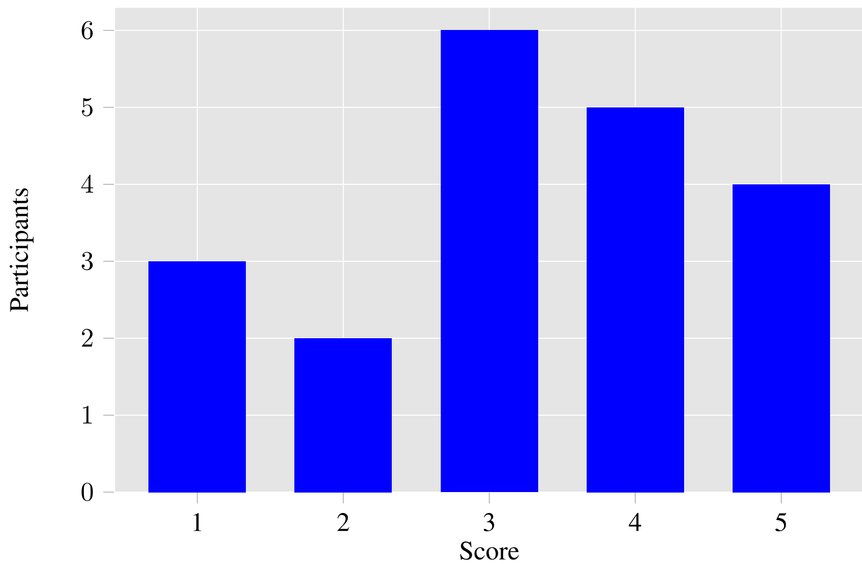


Figure 6.5:
Questionnaire statement:
I felt that a challenge for my skill level was available

6.2.3 Does the zoom functionality help the user to be more accurate?

A request from users in the earlier works was the ability to zoom on pictures. To make the application as great as it needs to be for the user to use it regularly, a good zoom feature was needed. In the current implementation, zoom is implemented on mobile devices with touch capabilities. While testing the application, the participants were requested to try the zoom functionality while performing the exercises in the outline mode. Making the user aware that the zoom functionality was an option made it possible to measure to what degree the users felt it helped them in finishing the outline exercises.

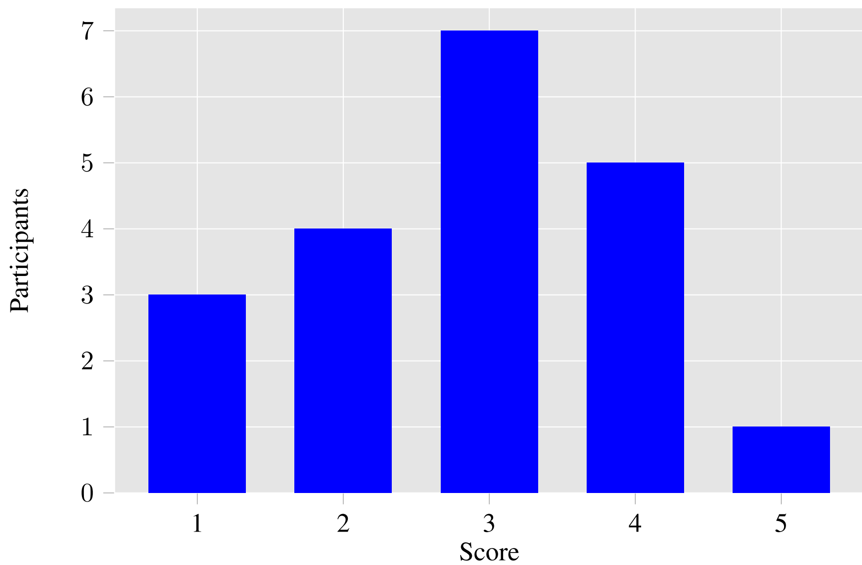


Figure 6.6:
Questionnaire statement:
The zoom functionality help me to be more accurate

The average answer for the statement, *The zoom functionality help me to be more accurate*, was 2.85, which was below average as shown in Figure 6.6. There were no definitive answers to why, but a few observations can be made. First, some users reported issues with their phone and the zoom functionality, which prevented them from using the feature and this would naturally produce a negative score. Using the feature was only encouraged, and not mandatory, which entails that users may have skipped trying it. In addition, there was no zoom available on the desktop version of the application, which means that users on this platform would not try the feature either. Users were encouraged to use a mobile device, but not forced. Using a tablet may also produce a picture with a adequate size to not need the zoom feature. Lastly, using the zoom feature may not have been as intuitive as excepted by the developers and users may have failed to figure out how it works.

6.3 Usability

As presented in Section 5.2.3, the usability test, System Usability Scale, were carried out with 45 participants. The overall score for the system was 72,84 which is considered to be an above average score. Although the application receive a lower usability score than the previous attempts, it is important to consider the significantly larger group of participants and that they are spread across different disiplines. Considering the textual feedback and the SUS score of 72,84, the application shows promise.

Chapter 7

Conclusion

In this thesis, the process of evaluating and developing existing and new solutions to enhance the available learning and teaching platforms available for medical imaging interpretation has been described and executed. During the process, an already existing platform has been expanded, and a new artifact has been developed. The goal of this research was to map how a learning tool with game elements could be used to encourage users to study more efficiently and how it could be perceived as a good learning tool. Two research questions were created to address these aspects of the goal.

- RQ1: How can we incentivize the use of this system?
  - Is the game fun to play?
  - Does scoring motivate the player?
- RQ2: To what extent is the application perceived as a useful learning tool?
  - Is the game perceived as a useful learning tool?
  - Does the application provide adequate challenge?
  - Does the zoom functionality help the users to be more accurate?

To answer the research questions a new application was developed. The new application focused on reusing the previous work as much as possible, while also contributing to the platform by adding better support for all platforms, especially mobile devices. Enhancing efficiency and ease for all users of the platform was a core principle through the development, and the previous work has seen improvements for administrators and experts allowing them to create task faster and easier than before.

RQ1: How can we incentivize the use of this system?

Analyzing the answers from the medical doctor students answering the questionnaire in the study, an indication can be seen that a lot of the participants is affected by gamification in a positive way as discussed in Section 6.1. When considering the overall feedback from the questionnaire, both statements and textual feedback, the results indicates that that game elements in a medical imaging interpretation application could be beneficial for the motivation of using such an application. The results indicate that the users would like more game elements to further increase the fun factor of the game. Comparing the answer given to *The game is fun to play* and *Getting a high score motivated me to play again*, it is possible that some players were more influenced by the game elements than they realized.

RQ2: To what extent is the application perceived as a useful learning tool?

The participants of the study indicated that an application of this sort would be of interest. The majority of the medical student gave positive answers on the statements regarding improving their skill by using the application, as discussed in Section 6.1. Indicating that the students believed that regular use of the application would be beneficial for them. Considering textual feedback, the main concern was the vast amount of exercise needed to be able to continuously learn new things from the application. Students also requested more information about each exercise in the application, indicating that there are improvements to be made, but a lot of potential. Some student would prefer more exercises on different skill levels, but the majority were also positive to the current implementation as a good starting point. The results gathered gives a strong indication that the users believed this application could improve or supplement their current study pattern, and the results indicates that the application has potential as a useful learning tool.

The final verdict indicates that further research and development should be performed in the direction of learning applications for medical imaging interpretation using game elements. The resulting platform supports experts, teachers, and administrators in exercise creation and development, and gives the user an easy way to access application that can be used from everywhere at any time.

Chapter 8

Future work

In this section improvements to the administrative interface will be suggested as well as suggested improvements to the mobile game. The improvements are based on the feedback given by test users as well as improvements to the performance based on tests done by the developers.

8.1 Using 3D Volumes

Creating medical imaging represented as 3D volumes, allows the user to explore the data by changing their view based on the axial, coronal or sagittal plane[1]. At St. Olavs 3D volumes are regularly created, and the solution for these scans are created alongside them.

From the developers perspective, it is firmly believed that this type of interactive exercises would be very interesting to the user as they provide a lot of information. They would give the users challenges presented in a similar format as in the real world. Considering that solutions are created with the scans will simplify the exercise generation process.

Using the raw data from scans opens up for other types of exercises as well. For example, a user can be asked to select an optimal frame from a scan based on ultrasound from an

[1] https://en.wikipedia.org/wiki/Anatomical_plane

approximate region. It would also be possible to create video exercises from the available scans. Unfortunately, this data was not available while writing this thesis.

8.2 Social features

Incorporating a global high score list was requested in the feedback from the user test. In the current implementation of the high score functionality all the data is store in the local database. An improvement to the system would be to connect the high score to a user account with a high score list stored in the backend. It would also be possible to integrate social media like Facebook and Tweeter to share the high score with friends.

8.3 Enhancements for the administration interface

This section contains suggested improvements to the administration interface.

8.3.1 Solution average

Calculating the average solution between outlines is implemented in the system. However, the calculation time is not optimal. In the current implementation, every solution added to the calculation will double the calculation time. A possible solution to this problem is to process multiple images simultaneously. By dividing the images into pairs and process them in parallel, the performance might be increased. The time complexity could potentially be reduced from $O(n^3)$ to $O(n \log n)$.

8.4 Enhancements for the game

This section suggests improvements for the game. The features described were given as feedback from users during the testing, suggested by experts and planned features that were not implemented due to time constraints.

8.4.1 Eraser function

An eraser function was requested for the Outline question. This feature would allow the user to enter an eraser mode where instead of drawing, they would remove parts of the outline. This feature is similar to the "Undo" feature already implemented, but can offer more accuracy.

8.4.2 Zoom button on all devices

The implemented zoom functionality currently only works on touch devices, as the desktop clients screen are larger in size and the canvas is rendered in an adequate size to not need this feature. However, the button for zoom mode is still visible on larger screens, and it should be reviewed if this functionality should be implemented or the button should be removed.

8.4.3 Scroll indications

In some parts of the interface, notably the answers to multiple choice questions, had a tendency to float out of the screen. To enhance the user experience, a good solution could be to indicate to the user that there are more options further down on the page. This could be done with a hovering arrow or some other scroll indicator like a scrollbar animation.

8.4.4 Button placement

Even though it is intended for the web application to be installed onto the home screens on smartphones, this is not mandatory. The user is free to use the application from the browser. As mentioned in Section 5.2.5, some user had trouble with the interface based on the fact that the browser address bar took up some of the screen real estate. When developing the application, this was not taken into account, and it may be necessary to relocate the buttons to make sure they are easily available.

8.4.5 User feedback

In the questionnaire, the test users asked for more feedback when they answer questions in the application. One way to do this would be to render the points achieved by a user after answering a question. There is a lot of animations which can be used to achieve this, similar to the high score view currently implemented.

8.4.6 Improve support for multiple areas on canvas

The support for adding multiple areas of interest in an outline and landmark exercise is available in the backend, but the implementation is not 100% complete in the application. Known shortages are that if the user enters a landmark question with two or more regions, the user can only place one mark. In the current implementation, the user will get a correct answer if they mark the correct area of *one* of the regions. This problem is similar in the outline mode of the application.

8.4.7 Annotations

Even though the system tries to implement the familiarity of using gestures like most mobile devices, some of the users still struggled to figure out how to zoom. A possible solution to this would be to, in addition to a guide on how to use the system, add annotations to the canvas and other parts of the applications. An example of annotations on a canvas, is Google Map's information on how to move around in the map on mobile devices shown in Figure 8.1.

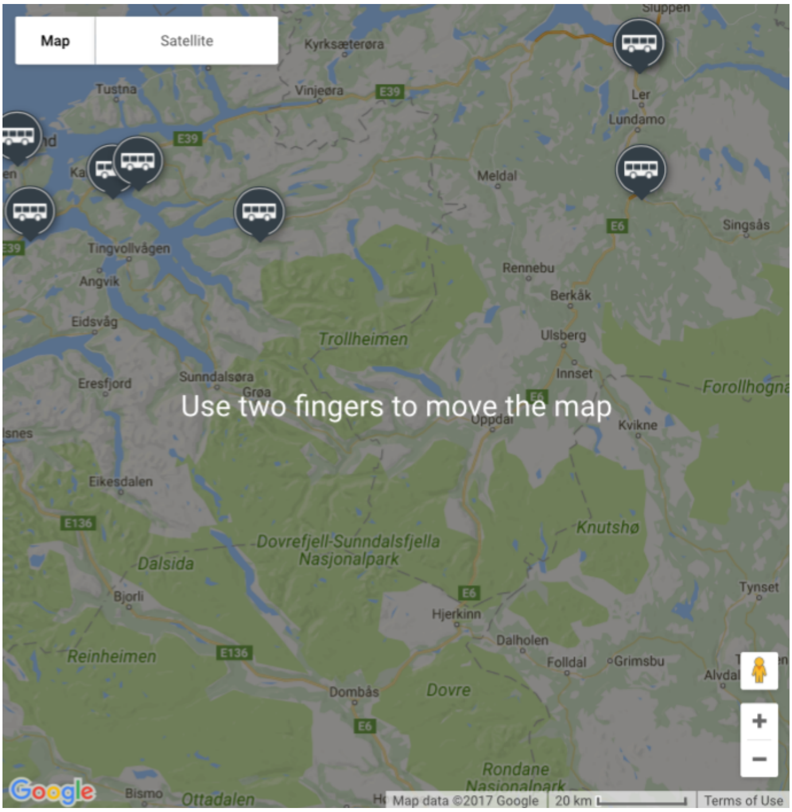


Figure 8.1:
Google maps on a mobile device

Bibliography

[1] S. J. Warden and J. M. McMeeken, "Ultrasound usage and dosage in sports physiotherapy," *Ultrasound in medicine & biology*, vol. 28, no. 8, pp. 1075–1080, 2002.

[2] P. Marhofer, M. Greher, and S. Kapral, "Ultrasound guidance in regional anaesthesia†," *BJA: British Journal of Anaesthesia*, vol. 94, no. 1, p. 7, 2005. [Online]. Available: +http://dx.doi.org/10.1093/bja/aei002

[3] J. T. Bushberg and J. M. Boone, *The essential physics of medical imaging*. Lippincott Williams & Wilkins, 2011.

[4] V. J. Shute, L. Rieber, and R. Van Eck, "Games… and… learning," *Trends and issues in instructional design and technology*, vol. 3, 2011.

[5] M. Csikszentmihalyi, "Flow: The psychology of optimal performance," *NY: Cambridge UniversityPress*, 1990.

[6] R. Garris, R. Ahlers, and J. E. Driskell, "Games, motivation, and learning: A research and practice model," *Simulation & Gaming*, vol. 33, no. 4, pp. 441–467, 2002. [Online]. Available: http://dx.doi.org/10.1177/1046878102238607

[7] D. I. Cordova and M. R. Lepper, "Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice." *Journal of educational psychology*, vol. 88, no. 4, p. 715, 1996.

[8] K. E. Ricci, E. Salas, and J. A. Cannon-Bowers, “Do computer-based games facilitate knowledge acquisition and retention?” *Military Psychology*, vol. 8, no. 4, p. 295, 1996.

[9] B. V. Whitehill and B. A. McDonald, “Improving learning persistence of military personnel by enhancing motivation in a technical trainiing program,” *Simulation & Gaming*, vol. 24, no. 3, pp. 294–313, 1993.

[10] D. Crookall and K. Arai, *Simulation and gaming across disciplines and cultures: ISAGA at a watershed.* Sage Publications, Inc, 1995.

[11] E. L. Deci and R. M. Ryan, “Intrinsic motivation and self-determination in human behavior. 1985,” *Consultado en septiembre*, 2013.

[12] E. A. Locke and G. P. Latham, *A theory of goal setting & task performance.* Prentice-Hall, Inc, 1990.

[13] K. L. Simons, “New technologies in simulation games,” *System Dynamics Review*, vol. 9, no. 2, pp. 135–152, 1993.

[14] A. Bandura and R. Wood, “Effect of perceived controllability and performance standards on self-regulation of complex decision making.” *Journal of personality and social psychology*, vol. 56, no. 5, p. 805, 1989.

[15] S. Hellan and H. H. Kamperud, “Teaching ultrasound-guided regional anesthesia through mobile phone games,” 2015.

[16] M. N. Hallan, “Medical imaging interpretation courseware,” 2016.

[17] M. S. Tønnessen, “A serious game for medical image interpretation,” 2016.

[18] W.-Y. Chan, J. Qin, Y.-P. Chui, and P.-A. Heng, “A serious game for learning ultrasound-guided needle placement skills,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 6, pp. 1032–1042, 2012.

[19] IRCAD, “Websurg,” 2017, last visited 2017-05-23. [Online]. Available: http://www.websurg.com/

[20] I. FUJIFILM SonoSite, "Sonoaccess 2.0 mobile app," 2017, last visited 2017-05-04. [Online]. Available: https://www.sonosite.com/education/sonoaccess-ultrasound-education-app

[21] S. Lee and J. W. Jeon, "Evaluating performance of android platform using native c for embedded systems," in *ICCAS 2010*, Oct 2010, pp. 1160–1163.

[22] C. M. Lin, J. H. Lin, C. R. Dow, and C. M. Wen, "Benchmark dalvik and native code for android system," in *2011 Second International Conference on Innovations in Bio-inspired Computing and Applications*, Dec 2011, pp. 320–323.

[23] T. M. Grønli, J. Hansen, G. Ghinea, and M. Younas, "Mobile application platform heterogeneity: Android vs windows phone vs ios vs firefox os," in *2014 IEEE 28th International Conference on Advanced Information Networking and Applications*, May 2014, pp. 635–641.

[24] B. Yee, D. Sehr, G. Dardyk, J. B. Chen, R. Muth, T. Ormandy, S. Okasaka, N. Narula, and N. Fullagar, "Native client: A sandbox for portable, untrusted x86 native code," in *2009 30th IEEE Symposium on Security and Privacy*, May 2009, pp. 79–93.

[25] H. Heitkötter, S. Hanschke, and T. A. Majchrzak, "Evaluating cross-platform development approaches for mobile applications," in *International Conference on Web Information Systems and Technologies*. Springer, 2012, pp. 120–138.

[26] T. Neil, *Mobile design pattern gallery: UI patterns for smartphone apps*. " O'Reilly Media, Inc.", 2014.

[27] E. ECMAScript, E. C. M. Association *et al.*, "Ecmascript language specification," 2011.

[28] D. Raggett, A. Le Hors, I. Jacobs *et al.*, "Html 4.01 specification," *W3C recommendation*, vol. 24, 1999.

[29] B. Bos, H. W. Lie, C. Lilley, and I. Jacobs, "Cascading style sheets, level 2 css2 specification," 1998.

[30] A. Charland and B. Leroux, "Mobile application development: web vs. native," *Communications of the ACM*, vol. 54, no. 5, pp. 49–53, 2011.

[31] StatCounter, "Statcounter globalstats," 2017, last visited 2017-05-20. [Online]. Available: http://gs.statcounter.com

[32] ——, "Statcounter factsheet," 2017, last visited 2017-05-20. [Online]. Available: http://gs.statcounter.com/factsheet

[33] J. Poushter, "Smartphone ownership and internet usage continues to climb in emerging economies," 2016, last visited 2017-05-20. [Online]. Available: http://www.pewglobal.org/2016/02/22/smartphone-ownership-and-internet-usage-continues-to-climb-in-emerging-economies/

[34] I. O. for Standardization, "Iso 9241 - 2010, ergonomics of human system interaction," 2010.

[35] S. L. Smith and J. N. Mosier, *Guidelines for designing user interface software*. Mitre Corporation Bedford, MA, 1986.

[36] J. Nielsen, "10 usability heuristics for user interface design," *Nielsen Norman Group*, vol. 1, no. 1, 1995.

[37] W. Lidwell, K. Holden, and J. Butler, *Universal principles of design, revised and updated: 125 ways to enhance usability, influence perception, increase appeal, make better design decisions, and teach through design*. Rockport Pub, 2010.

[38] E. Marcotte, *Responsive web design*. Editions Eyrolles, 2013, no. 4.

[39] G. van Rossum and P. S. Foundation, "Python," 2016. [Online]. Available: https://python.org

[40] C. J. Date and H. Darwen, *A Guide to the SQL Standard*. Addison-Wesley New York, 1987, vol. 3.

[41] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, "Extensible markup language (xml)." *World Wide Web Journal*, vol. 2, no. 4, pp. 27–66, 1997.

[42] T. Bray, "The javascript object notation (json) data interchange format," 2014.

[43] E. Czaplicki, "Elm language," 2016. [Online]. Available: http://elm-lang.org/

[44] R. Milner, *The definition of standard ML: revised*. MIT press, 1997.

[45] S. Owens, "A sound semantics for ocaml light," in *European Symposium on Programming*. Springer, 2008, pp. 1–15.

[46] S. Marlow *et al.*, "Haskell 2010 language report," *Available online http://www. haskell. org/(May 2011)*, 2010.

[47] IBM, "Three-tier architectures," 2017, last visited 2017-03-14. [Online]. Available: https://www.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/covr_3-tier.html

[48] D. Barry and T. Stanienda, "Solving the java object storage problem," *Computer*, vol. 31, no. 11, pp. 33–40, 1998.

[49] M. Masse, *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*. " O'Reilly Media, Inc.", 2011.

[50] D. Schepers, S. Moon, M. Brubeck, and A. Barstow, "Touch events version 1," 2013, last visited 2017-03-18. [Online]. Available: https://www.w3.org/TR/2013/PR-touch-events-20130509

[51] T. Boutell, "Png (portable network graphics) specification version 1.0," 1997.

[52] U. D. of Health Human Services, "System usability scale (sus)," 2017, last visited 2017-05-20. [Online]. Available: https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html

[53] J. Nielsen, "How many test users in a usability study?" 2012, last visited 2017-05-21. [Online]. Available: https://www.nngroup.com/articles/how-many-test-users/

[54] J. Butter, T. H. Grant, M. Egan, M. Kaye, D. B. Wayne, V. Carrión-Carire, and W. C. McGaghie, "Does ultrasound training boost year 1 medical student competence

and confidence when learning abdominal examination?" *Medical education*, vol. 41, no. 9, pp. 843–848, 2007.

Appendix A

Euclidean average comparison

A.1 Case 2

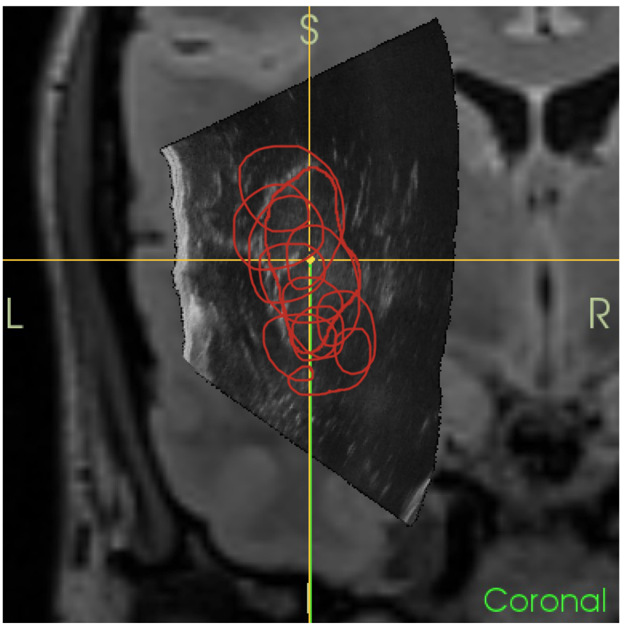


Figure A.1:
Rendered image with solution and user input from Case 2

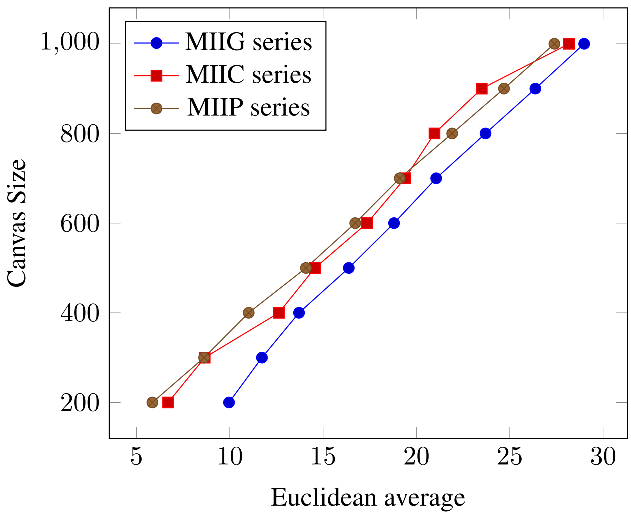


Figure A.2:
Plotted result from each Euclidean average algorithm on different resolution from Case 2

Size	MIIG	MIIC	MIIP
1000	28.987238165080935	28.174660634889744	27.393873246195763
900	26.373915081150262	23.501809467682577	24.693113896505935
800	23.703202216489768	20.968493654972484	21.916799755975216
700	21.0593270069141	19.38946303913566	19.107981223494033
600	18.80187411762266	17.359298339811378	16.722344299762437
500	16.377630588271813	14.562701856141404	14.082009353881055
400	13.711239070025627	12.632754447657991	11.015557630637247
300	11.724774697908037	8.658606679636732	8.625477468701412
200	9.960377885745668	6.699564631948674	5.85683268771412

Figure A.3:
Results from each Euclidean average algorithm on different resolution from Case 2

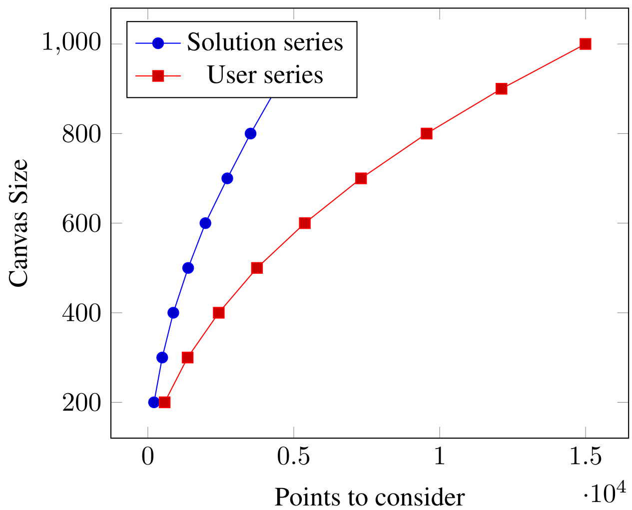


Figure A.4:
Plotted amount of user submitted pixels and pixels in solution from Case 2

Size	Solution	User
1000	5478	14995
900	4395	12118
800	3521	9553
700	2724	7306
600	1974	5380
500	1382	3741
400	874	2431
300	494	1364
200	213	577

Figure A.5:
Amount of points to consider from the solution and the user given a resolution from Case 2

A.2 Case 3

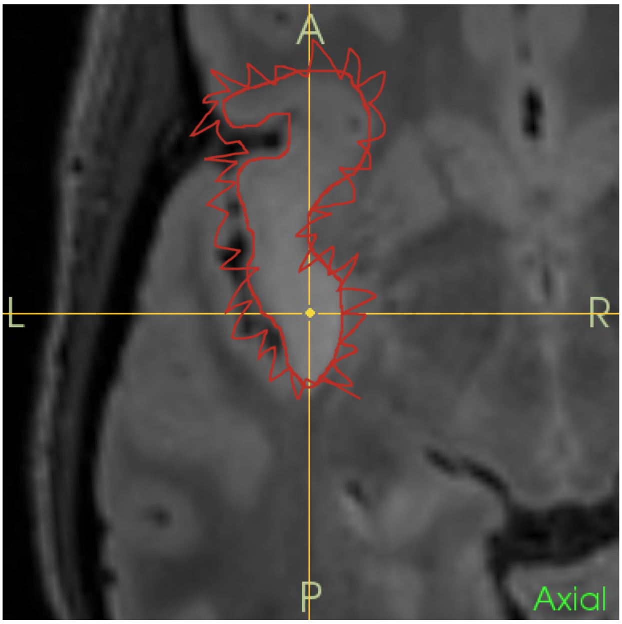


Figure A.6:
Rendered image with solution and user input from Case 3

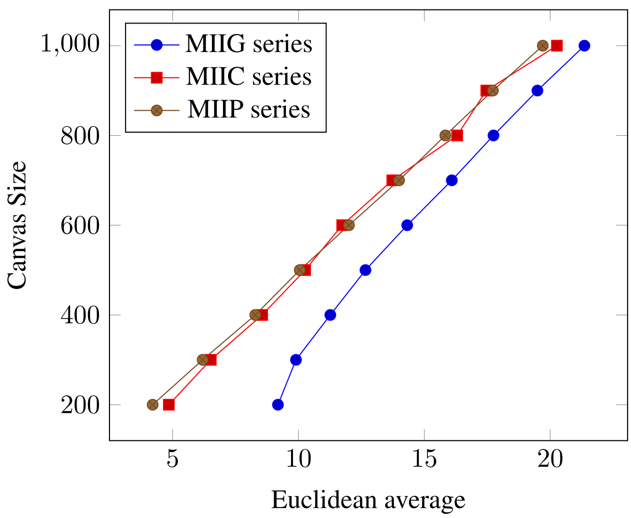


Figure A.7:
Plotted result from each Euclidean average algorithm on different resolution from Case 3

Size	MIIG	MIIC	MIIP
1000	21.359933299941535	20.267531403548052	19.70587213513538
900	19.493280103068088	17.464129554620328	17.723999449530254
800	17.74830632118026	16.305448439574562	15.834896050605153
700	16.092326617004392	13.728088334997773	13.996301043377755
600	14.319529523738721	11.737098232038937	12.00540404700303
500	12.661793313342834	10.262051595278583	10.050849205995911
400	11.26469234340745	8.5574271099827	8.283675036960135
300	9.900598024221875	6.51458944193373	6.191669217448087
200	9.184065252618725	4.849622327629818	4.20639652697487

Figure A.8:
Results from each Euclidean average algorithm on different resolution from Case 3

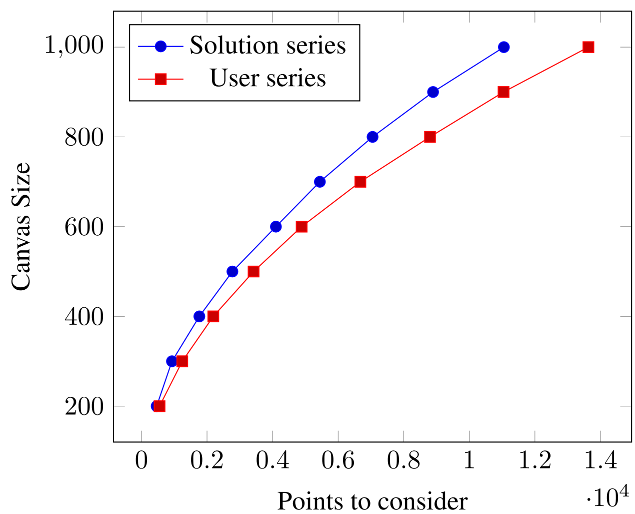


Figure A.9:
Plotted amount of user submitted pixels and pixels in solution from Case 3

Size	Solution	User
1000	11054	13632
900	8894	11044
800	7049	8801
700	5442	6677
600	4101	4886
500	2774	3420
400	1766	2192
300	931	1245
200	464	552

Figure A.10:
Amount of points to consider from the solution and the user given a resolution from Case 3

A.3 Case 4

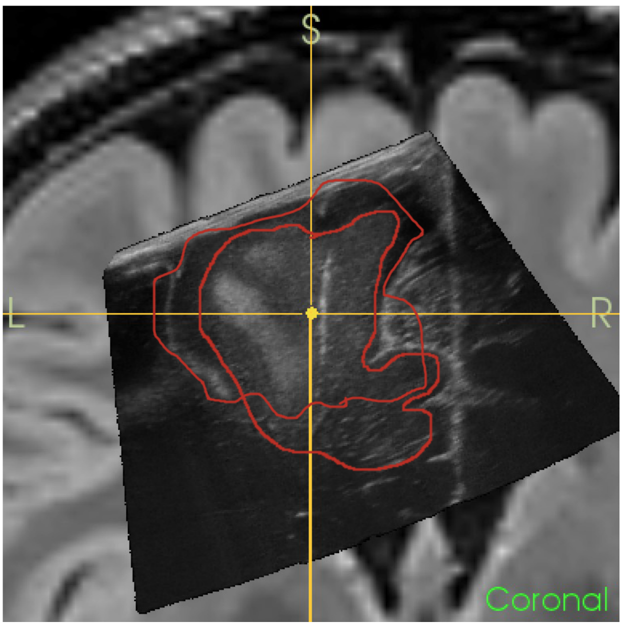


Figure A.11:
Rendered image with solution and user input from Case 4

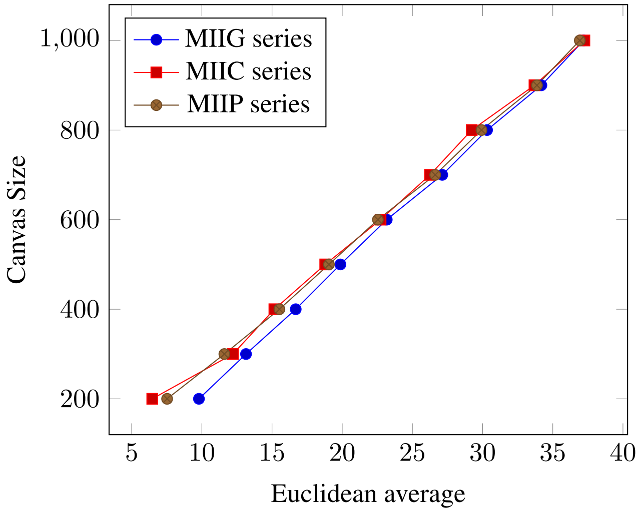


Figure A.12:
Plotted result from each Euclidean average algorithm on different resolution from Case 4

Size	MIIG	MIIC	MIIP
1000	37.19866453713232	37.24996685546179	36.926122921486666
900	34.18824382669349	33.697324595089796	33.87220464623932
800	30.318335571211207	29.223928227937378	29.923300821506444
700	27.12667807844869	26.257140736001535	26.6227257283155
600	23.17094782748622	22.73125476448451	22.54268717316835
500	19.87171479713707	18.789911993717276	19.053947653958623
400	16.687120328654785	15.16781491102303	15.525746693387257
300	13.145754878574756	12.21545572749468	11.604902554625637
200	9.792354786249337	6.468621371063862	7.5254641294252425

Figure A.13:
Results from each Euclidean average algorithm on different resolution from Case 4

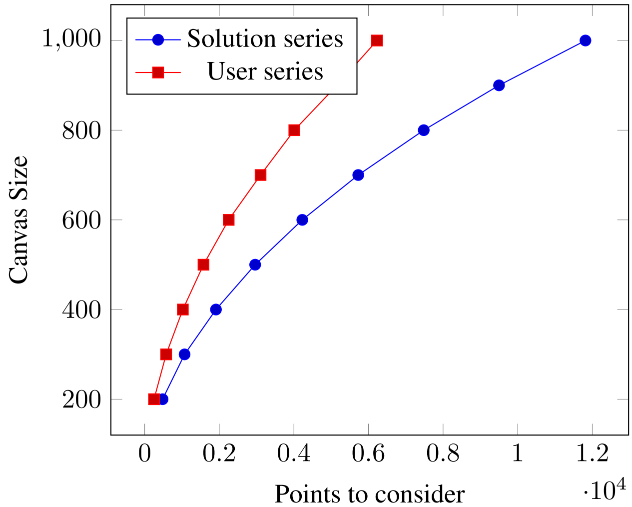


Figure A.14:
Plotted amount of user submitted pixels and pixels in solution from Case 4

Size	Solution	User
1000	11817	6229
900	9500	5155
800	7482	4012
700	5726	3107
600	4226	2252
500	2962	1577
400	1911	1023
300	1070	576
200	480	251

Figure A.15:
Amount of points to consider from the solution and the user given a resolution from Case 4

Appendix B

Questionnaire

MIIG - questionnaire

Instructions: For each of the following statements, mark one box that best describes your reactions to the application today.

* Required

1. **1. What is your medical background? ***

   Mark only one oval.

   - Medical doctor
   - Medical doctor (student)
   - Other healthcare professional
   - Other healthcare professional (student)
   - Technologist
   - Technologist (student)
   - Other profession
   - Other profession (student)

2. **2. I have self interest in learning medical imaging interpretation ***

   Mark only one oval.

	1	2	3	4	5	
Strongly disagree						Strongly agree

3. **3. I think that I would like to use this application frequently. ***

   Mark only one oval.

	1	2	3	4	5	
Strongly disagree						Strongly agree

4. **4. I found this application unnecessarily complex. ***

   Mark only one oval.

	1	2	3	4	5	
Strongly disagree						Strongly agree

5. **5. I thought this application was easy to use.** *
Mark only one oval.

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

6. **6. I think that I would need assistance to be able to use this application** *
Mark only one oval.

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

7. **7. I found the various functions in this application were well integrated.** *
Mark only one oval.

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

8. **8. I thought there was too much inconsistency in this application.** *
Mark only one oval.

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

9. **9. I would imagine that most people would learn to use this application very quickly.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

10. **10. I found this application very cumbersome/awkward to use** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

11. **11 . I felt very confident using this application.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

12. **12. I needed to learn a lot of things before I could get going with this application.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

13. **13. I am familiar with interpretation of Ultrasound images** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

14. **14. I am familiar with interpretation of MR images** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

15. **15. The game is fun to play** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

16. **16. Getting a high score motivated me to play again** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

17. **17. The zoom functionality help me to be more accurate** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

18. **18. Regular use of this application will improve my medical imaging interpretation skill** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

19. **19. I felt that a challenge for my skill level was available** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

20. **20. It would be beneficial for my current medical imaging interpretation skill to include this application in my training** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

21. **21. I would recommend this app to a friend** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

22. **22. Is there something you think could improve the application?**

Appendix C

Usertest

User test for the MIIP application

Navigate to this url:

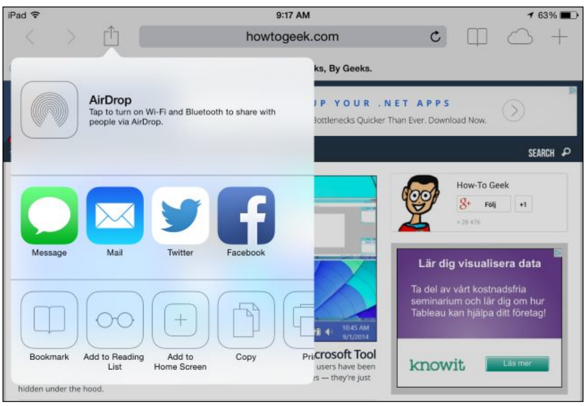
miip.idi.ntnu.no/app

On your mobile device (smartphone or tablet) add the webpage to your homescreen. Guide for add to home screen for iOS and Android below.

How to install application

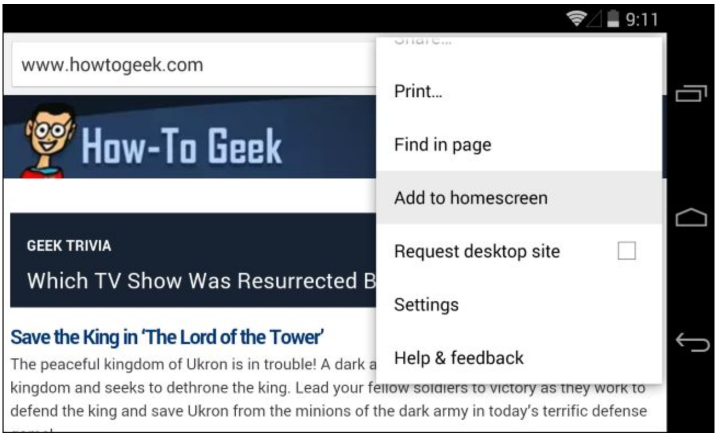
iPhone, iPad, & iPod Touch

Launch the Safari browser on Apple's iOS and navigate to the website or web page you want to add to your home screen. Tap the Share button on the browser's toolbar — that's the rectangle with an arrow pointing upward. It's on the bar at the top of the screen on an iPad, and on the bar at the bottom of the screen on an iPhone or iPod Touch. Tap the Add to Home Screen icon in the Share menu.



Android

Launch Chrome for Android and open the website or web page you want to pin to your home screen. Tap the menu button and tap Add to homescreen. You'll be able to enter a name for the shortcut and then Chrome will add it to your home screen.



Exercises user test.

Choose the “Multiple Choice Questions” category.

- Choose to answer 5 questions.
- Answer all the questions.
- Return to the home screen of the application.

Choose the “Landmark Questions” category.

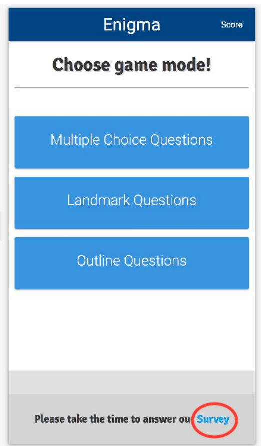
- Choose to answer 2 questions.
- Answer all the questions.
- Return to the home screen of the application.

Choose the “Outline Questions” category.

- Choose to answer 4 questions.
- Zoom in on the picture.
- Answer all the questions.
- Return to the home screen of the application.

Click the score “Score” option to see statistics.

- What is your total high score?



When this is done please take the time to answer our survey.
The survey is available here: Survey
https://goo.gl/forms/8F67T2z8XC0P1c362
Or at the bottom of the application: