Bachelor's thesis

Information and Communications Technology

2020

Oskari Rintamäki

GAMIFYING THE TEACHING OF NEUROBIOLOGY AND EEG BASICS

TURKU AMK
TURKU UNIVERSITY OF
APPLIED SCIENCES

BACHELOR’S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information and Communications Technology

2020 | 43 pages

Oskari Rintamäki

GAMIFYING THE TEACHING OF NEUROBIOLOGY AND EEG BASICS

Teaching and demonstrating the basics of neurobiology and the function of electroencephalography (EEG) is a very hard task. Up until the last few years, the teaching of these subjects involved the theoretical literature and two-dimensional images at our disposal and, at most we have had some educational videos or physical models to help with learning.

This thesis is a portfolio documenting the developing of an application for Android platform, that visualizes the subject matter by using AR technology and is easy to integrate in teaching. The goal was to utilize augmented reality technology to display and simulate 3D models. The end result had to be a working application that is not only entertaining, but also aids the learning process through the interactive experience.

Initially, the design of the application required sufficient knowledge of the subject to be gamified. After the required knowledge base, the contents of the application were planned together with the client. The required elements were modeled using 3D modeling software Blender. The designed 3D models were combined with the material into an application created with the game engine Unity. A major part of the application development was the user interface (UI) and general user experience (UX), as well as animating the created models correctly based on the subject material.

The work resulted in the development of an application that competently simulates a number of topics in neurobiology and EEG. The application was designed to be easy to use but also versatile, and its appearance was in line with the theme of the subject matter in a modern way.

KEYWORDS:

EEG, gamification, mobile application, augmented reality

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tieto- ja viestintätekniikka

2020 | 43 sivua

Oskari Rintamäki

NEUROTIETEIDEN JA EEG:N PERUSTEIDEN OPETUKSEN PELILLISTÄMINEN

Neurobiologian ja EEG:n toiminnallisuuden perusteiden opiskelu ja havainnointi on hyvin vaikeaa. Tähän asti käytössä on enimmäkseen ollut vain teoriatekstit sekä kaksiulotteisia havainnollistavia kuvia. Joskus opetuksessa voidaan hyödyntää myös videomateriaalia tai fyysisiä malleja.

Tämä opinnäytetyö on portfolio, joka esittelee mobiilisovelluksen suunnittelun ja toteutuksen Android-käyttöliittymälle. Sovelluksen tarkoitus on havainnollistaa vaadittuja aiheita käyttäen AR-teknologiaa, sekä olla helposti integroitavissa opetukseen. Tavoitteena oli hyödyntää lisättyä todellisuutta kolmiulotteisten mallien esittelemiseen ja käsittelyyn. Lopputuloksena haluttiin kokonaisuus, joka pelillistää viihdyttävästi opetettavia aiheita, mutta samalla auttaa teorian sisäistämistä interaktiivisen kokemuksen kautta.

Aluksi sovelluksen suunnitteluun vaadittiin riittävä tietämys pelillistettävästä aiheesta. Tarvittavan tietoperustan jälkeen suunniteltiin toimeksiantajan kanssa yhdessä, mitä elementtejä sovellukseen haluttiin sisällyttää. Vaaditut elementit mallinnettiin käyttämällä Blender-ohjelmistoa. Suunnitellut 3D-mallit koottiin yhteen materiaalin kanssa sovellukseen, joka luotiin Unity-pelimoottorilla. Isossa osassa kehitystyötä oli sovelluksen ulkoasu ja käyttäjäkokemus, sekä luotujen mallien animointi vastaamaan taustamateriaalia.

Työssä saatiin kehitettyä sovellus, joka simuloi pätevästi useita taustamateriaalin käsittelemiä aiheita. Sovelluksesta saatiin helppokäyttöinen, mutta monipuolinen, ja sen ulkoasu vastasi käsiteltyjen aiheiden teemaa modernilla tavalla.

ASIASANAT:

EEG, pelillistäminen, mobiilisovellus, lisätty todellisuus

CONTENTS

FIGURES

LIST OF ABBREVIATIONS

AR	Augmented Reality
Blender	Free and open-source 3D computer graphics software toolset
C#	Object oriented programming language
Debugging	Finding issues with written code
Dipole	A region of positive charge separated from a region of negative charge by some distance.
Illustrator	Adobe Illustrator
Mesh	Collection of vertices, edges and faces that make the 3D object
QR code	Quick response code.
Sprite	2D image used in UI elements
Unity	Free to use game engine used for developing 2D and 3D games
Vuforia	AR software development kit (SDK) for mobile devices

1 INTRODUCTION

Virtual and augmented reality (VR / AR) are exponentially evolving technologies that have yet to see their full potential. Each year, these technologies evolve and new products come to the market taking advantage of the new functionalities. Initially, virtual and augmented reality were created for private users solely for entertainment purposes, mainly for video games. Since then, they have begun to be more widely used in different industries for different purposes like education, marketing and product development.

The ease of use, availability and lower cost of VR / AR technologies have made it possible to utilize them in applications that can run even with just a modern smartphone. This creates new opportunities for teaching many subjects that would often require a great number of expensive equipment or it would be very difficult to obtain material like microscopes and biological material. This is the basis for the decision to create an augmented reality application that gamifies the teaching of the fundamentals of electroencephalopgraphy (EEG) and neurobiology.

The function of EEG is based on alternating charges, potentials created by synapses, in different parts of the brain. Electrodes are placed in specific areas on the patient's head, which then register variations in these charges and differences between areas of the cortex and cranial surface (Olejniczak 2006). It is very difficult to demonstrate the theory of such a subject, since these events are not possible, or very hard, to film or capture in any way. All that remains are the illustrative pictures and videos created using 3D modeling. However, these are often inadequate and difficult to read.

The purpose of this thesis is to create and present a mobile application that solves the above problems. When finished, the application should be able to visualize the activity of brain neurons in a clear and entertaining way and aid future users in learning the subjects at matter.

The background work focuses on the central elements of brain neuronal function and the theory behind the workings of EEG. The purpose of the background work is to create a solid knowledge base on which the application can be built to be reliable and accurate. This includes exploring the various software and technologies needed to make well-informed decisions in the application development process.

In this thesis, the application is designed and implemented using the knowledge base of the background work and previously acquired knowledge on game and software development. There are many aspects to the development of the application, such as 3D modeling, UI and UX design, and programming in C#. The modeling must be as accurate as possible and designed so that it can be animated to simulate different functions associated to EEG’s working principles. The interface must reflect the topics covered, and be clear and intuitive. The application's target platform is the Android operating system.

The second chapter of this thesis goes trough the technologies and ways AR is and has been used. Chapter three provides the basic understanding and theory behind the inner workings of EEG. Chaper four explains what gamification is and how it is applied to this application. In the fifth chapter the planning process of the application is explained with further research on each part of application development process. The sixth chapter shows the results of the finished application with pictures and by explaining further how the application was made in the end, was the plan followed and what changed during the development. Chapter seven gives a use case for the application, going trough how the end user will use it. The final chapter, chapter eight, is for reflecting on the application development process and how the application could be developed further, what could have been done better and what to do in the future.

2 AR IN SOFTWARE DEVELOPMENT

Augmented reality has a long history, although today's modern solutions are quite new. Augmented reality, in all its simplicity, is the interaction of the real world and the virtual world, generally adding virtual content to the real world through an image viewed through a device (Merriam-Webster 2019). This virtual content, like 3D models, can be completely realistic and the user can interact with them in real time. The first mobile application using modern AR technology was created in 2004 (Karhu 2013).

The AR market is now worth more than € 3 billion globally, with expected amount of billion users by the year 2020. Many use AR applications every single day without even realizing it. The most trendy and popular AR applications at this time are Snapchat and Instagram with over 1.2 billion active users (Stat 2019). Particularly, one of Snapchat's greatest appeals is to take pictures and videos through a variety of filters that add objects to the real world, usually changing the look of the user's head area, especially the face. Due to the augmented reality's usability and sufficient versatility, it is believed to exceed the global revenue generated by virtual reality by 2020 (Moss 2019).

Thus, it can be stated that augmented reality is the prevailing technology of today's and future market, at least by being integrated in certain features of different applications. Its exponential development is supported by leaps in the technological advancements of mobile devices and the popularity of social media, as well as technology intelligence of younger generations.

2.1 Different forms of AR

Augmented reality can be utilized in many different ways, on many different devices. In particular, over the last ten years, a variety of devices has been developed to optimize the use of augmented reality, particularly for industrial use. These have often been different types of wearable glasses, through which one or both lenses allow the user to view virtual content almost in the same way as holograms from science fiction films. Unfortunately, only a few of these have seen success because the technologies used have not been advanced adequately and the costs of the equipment have been too high (Wheeler 2019).

Arguably the most successful form of AR is mobile applications, which bring a varying amount of augmented reality to the user. Modern smartphones make augmented reality accessible to everyone with no extra cost, and developed applications work the same way on all supported devices, the only difference being the operating system (Android / iOS).

2.2 The possibilities and limitations of AR

Augmented reality creates ample opportunities for the development of various applications. Developers can create virtual reality content at a low cost to the developer and the consumer. Augmented reality requires only a mediocre smartphone, while VR content requires special VR glasses and a fairly powerful computer. This makes AR easily approachable. This is why AR is selected for this application. AR can be used in everything from education to pure entertainment.

AR also has many limitations with current state of technology. The more complicated the AR portion of the application is, the more processing power it requires from the device. Older devices may be compatible with multiple or almost all AR applications, but the application may be slow and work poorly. Not all devices support projecting the content into air or a plane without an image target, so that means there has to be a picture that the user targets with their camera to get the content of the application to show. Having an image target is not always possible and it has to be taken into account when developing the application. Compared to VR, AR is also more constricted considering the interaction between the user and the content. AR applications don’t have the “virtual hands” to interact with content like in VR applications. AR supports some virtual buttons that the users can press, but those do not currently work very well and are mostly considered a gimmick.

3 INTRODUCTION TO EEG

Electroencephalography (EEG) is a “noninvasive test that records electrical patterns in your brain” (Mayfield Brain & Spine 2020). When measuring EEG, electrodes are placed on the patient's head, most of the time using a special headgear. This headset has ready places for the electrodes as they need to be located at certain positions on the head in order to efficiently and reliably record electrical activity. The nerve cells in our brains are creating small electrical signals, which then form brain waves that the electrodes can record (Mayfield Brain & Spine 2020).

The primary source of EEG signal comes from the certain way that cortical neurons line up in our brains. When these neurons group up in a way that their dipoles sum together, the electrodes are able to record the activity. This means the neurons have to be grouped in a way that their positive and negative dipoles are located next to each other respectively. Otherwise the charges cancel each other out and the electrical activity is not strong enough to be recorded on the scalp. In addition to having to be located parallel to each other, the neurons must be synchronously active and produce large enough signal (Olejniczak 2006).

3.1 Teaching of neurobiology

The basics of neurobiology are studied in some form in highschool biology courses and in several degree programmes in universities. The target userbase of this application is the students of these biology courses. It is intended to be used inside and outside of class, depending on the curriculum and preferences of the teacher. The aim of the application is to support the student in their studies, to aid the more visual learners who might not completely understand the subjects from just reading learning material. In general, visual aids are important in learning in addition to oral presentation (Zandan 2020).

Visual aids, in biology courses in general, are lacking and mostly outdated. They consist mostly from learning videos, pictures and sometimes physical models. There is a need for modern effective ways to present neuron level events. This application takes the learning material and makes it come alive with AR technology. Gamifying the otherwise monotonous learning helps the students stay interested and engaged in the subject at hand. This could be achieved similarly with the physical models, but those are expensive and take a lot of space when stored and in a classroom. Physical models also lack the ability to recreate events like this application does.

4 GAMIFICATION

Gamification is the process of inserting gameplay-like elements to products or services to enhance user engagement, and to motivate participation and loyalty. Gamification is taking advantage of already known techniques game designers use to motivate users to continue playing and feel accomplished. This often includes reward systems, leaderboards, and generally fun interaction with the product or service. Gamification is often used as a tool to increase performance, prevent boredom of mundane tasks, and to generate more interest and get larger userbase (Growth Engineering 2020).

Even though gamification usually includes reward systems and leaderboards, those are not required. The most important part of gamification is engagement and interaction. Reward systems and leaderboards are easy way to make the users feel like they achieved something and feel rewarded instantly, but they also are often unnecessary and even harmful depending on the service or product. While leaderboards reward users for doing good in something, they may also cause unhealthy competition and make users not in the top feel worse. Reward systems like gaining achievements or points are good, but may not be necessary and can also cause issues just like leaderboards. We can take Instagram as an example. Instagram is somewhat gamified because of the likes the users can give and get. Receiving more likes can feel good and rewarding, but then people can start comparing themselves to creators that receive more likes than them.

Gamification is often compared to learning games, which technically can be classified as gamified learning, but gamification does not equal a game. When gamifying a product or service, we take only the necessary elements of an actual game that can enhance the product and make it more fun. This often means visualizing something, creating animations or interactable objects. Instagram and Snapchat use AR to further gamify their services by offering different interactive “lenses” or filters that users can add to videos and pictures. Learning games are something between a game and gamification. Making a game for training certain part of a job can be considered gamifying. We can make games that allow employees to train dangerous or difficult jobs without endangering anyone or having to spend major resources.

When gamifying something we have to remember that we are not making a game. There has to be a purpose for it in achieving something, in solving a problem or learning something new. Gamification has to improve the product or service and not just add something to it. It can create healthy competition, improve learning, grow the userbase, and much more.

5 DESIGN PROCESS OF THE APPLICATION

After getting the necessary knowledge base, application development continued with a complete plan of how it should be implemented in the application and what the most important technologies were. A good plan and design will prevent major problems and give a clear direction for the actual development process, saving time and money (Murphy 2019).

Development process included three roles:

1. developer
2. client
3. user

The developer role included everything from consept design to implementation (modeling, coding etc.). The role of the developer also included the normal roles of project manager, project owner, software tester and so on. The client acted as both client, and the user. In addition to these roles there was a thesis instructor who provided support trough every part of the development process as needed. Both client and the instructor were closely involved in the design process of this application.

This application was designed and developed following user-centered design (UCD) and participatory design principles (Elizarova, Briselli and Dowd 2020). These design priciples include different roles in the design and overall development process and emphasize the role of the actual user. User and client participation is emphasized in every part of the development (Casali 2020). This ensured that the created application included all the key elements and they were based on correct information. The client could offer their expertise considering the content of the application, and provide some material for it. The client was a key part of the development process from beginning to end, so they could point out any flaws in the application or provide insight on what to add to it. They could verify the validity of each part of the application from the perspective of both the client and the user.

The development process was started by mapping the needs of the application.The materials and general content of the application were discussed and agreed on with the client. The decision was based on what can be included in the application within the given time and other resources. It is very important to be honest and clear about what is realistically possible and what challenges may arise in the development process so that expectations are not too high and work can be carried out as agreed.

The purpose of the application was to gamify the teaching of EEG’s theory and at the same time teach the fundamentals of neurosciences, which are important for understanding the function of EEG. The application sought to include textual material and simulations showing how various events work in practice at the neuronal level. This required the source material to be delivered to me by the client, which was then utilized in creating the 3D models and simulations.

5.1 Selected software

Android was the chosen operating system because iOS development requires products from the Apple family, which were not available for the development. In addition to that, Android is The biggest factor in the selection process was previous experience in product development. Since the author already had several years of experience with certain software, it was not necessary to learn new ones. In addition, there was two other constraints, budget and time. The entire application was aimed to be developed with a near-zero budget, which was a voluntarily intentional challenge. The selected software had to work seamlessly together to ensure a smooth development process and workflow.

Unity (v2018.2.20f1) was selected as the game engine, with the strengths of ease of use, accessibility and flexibility. Unity comes with a number of free add-ons and is quite compatible with other software. Unity's programming language is C # and it has a built-in animation tool that was used to animate simulations. It also has great tools for building an intuitive and modern user interface. Unity is versatile with a graphical user interface that saves time and effort in programming. As it’s not necessary to write out all the simplest and mundane code by hand, it enables the developer to allocate that saved time to other more complicated and important tasks.

Vuforia (v8.0.10) was selected as the platform for augmented reality in the application. Vuforia is a partner of Unity and so it’s also free to use with certain restrictions. Vuforia is a simple to use platform and supports a wide range of mobile devices. Thanks to their collaboration, Unity and Vuforia work seamlessly together, which is another risk factor removed from the development process.

Blender (v2.78) was the chosen software for 3D modeling. Blender is a free open-source toolset for creating 3D models, animations, and much more. Like Unity, it is also very easy to use and the author already had previous experience with it. Blender is very compatible with Unity and also includes a built-in tool for animating objects if Unity's own tool was not enough.

Adobe Illustrator and Adobe Photoshop were the chosen software to create the user interface's (UI) graphic elements. Illustrator is a particularly good tool for creating UI elements, as it uses vector based graphics instead of pixels. This makes it easy to edit graphical elements as needed without having to worry about pixels before transferring them to Unity. These graphics can be for example resized and they look exactly the same, retaining its sharpness unlike pixel graphics that eventually get blurry.

5.2 UI/UX design plan

The basic structure of the application is very simple and straight forward. It has three parts:

1. Game scene
2. Material scene
3. Menus.

The structure of the application was kept simple to ensure a good user experience (UX). Because of the complexity of the topic being taught, the application had to be intuitive and easy to use. Buttons have to be easy to see and look like actual buttons. They have to be located where the user would expect them to be and they need to labelled correctly so the user knows what they do. Generally, the fewer button presses needed for the user to reach the content they want to reach, the better the user experience (Babich 2020). Therefore, from each scene, as shown in Figure 1, the user is provided direct access to the main menu and the main menu provides direct access to the core functionality of the application, i.e. the Game scene.

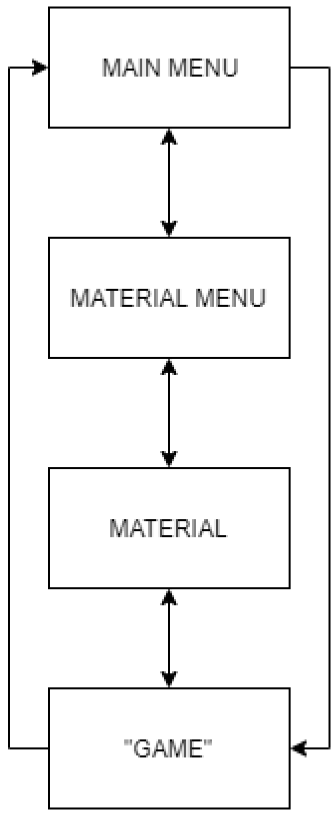

Figure 1. Basic structure of the application.

UX design should consider why and how to use the product (Inter-action Design Foundation 2019). The key to everything is intuition. The user should be able to operate the product instinctively without specific instructions. This is achieved by keeping the UI simple, easy to read, and the symbols used should be understandable regardless of the user's culture, language, and background (Vox 2018). Thus, the application sought to use symbols as universal as possible, following the most common standards in application development. Some examples are the question mark and the gear symbol used to describe the "help" section and settings. Although the work did not fully comply with all the Android standards, references were searched and the development was aimed to be as close to industry standards as possible.

5.2.1 UI design plan

The UI was designed to correspond to the theme of the application, the neurosciences. Neurosciences are part of biology and medicine, according to which the first and one of the most important parts of UI design, the color scheme, was designed. Words such as cleanliness, water and nature are associated with medicine. The most common colors associated with these things are white, blue and green (99designs 2019). This resulted in an almost monochromatic color scheme consisting mainly of different shades of blue and white, as seen in Figure 2.

Figure 2. Color scheme of the application.

I wanted the app to be modern and approachable. Because the purpose of the work was gamification, the look also had to be a bit playful and fun. Recent trends favor rounder shapes rather than sharp angles, as well as a certain amount of minimalism and color gradient (Fabian 2019). This approach was also consistent with the general theme, as continuous soft and rounded shapes are more natural than sharp right angles, and minimalism focuses the user's attention on important parts of the application. Generally, perfect sharp angles combined with the clinical color scheme also signal a certain seriousness that was avoided.

The main menu should be inviting and immediately highlight the mood and theme of the app. This is the user's first touch to the application, and a poorly designed menu may cause the user to not want to use the application. The main menu must be clear and tell the user immediately how to navigate to each part of the application. In general, the main menu has a button for the game, settings, instructions and background material. Below in Figure 3 is shown the planned structure of the main menu.

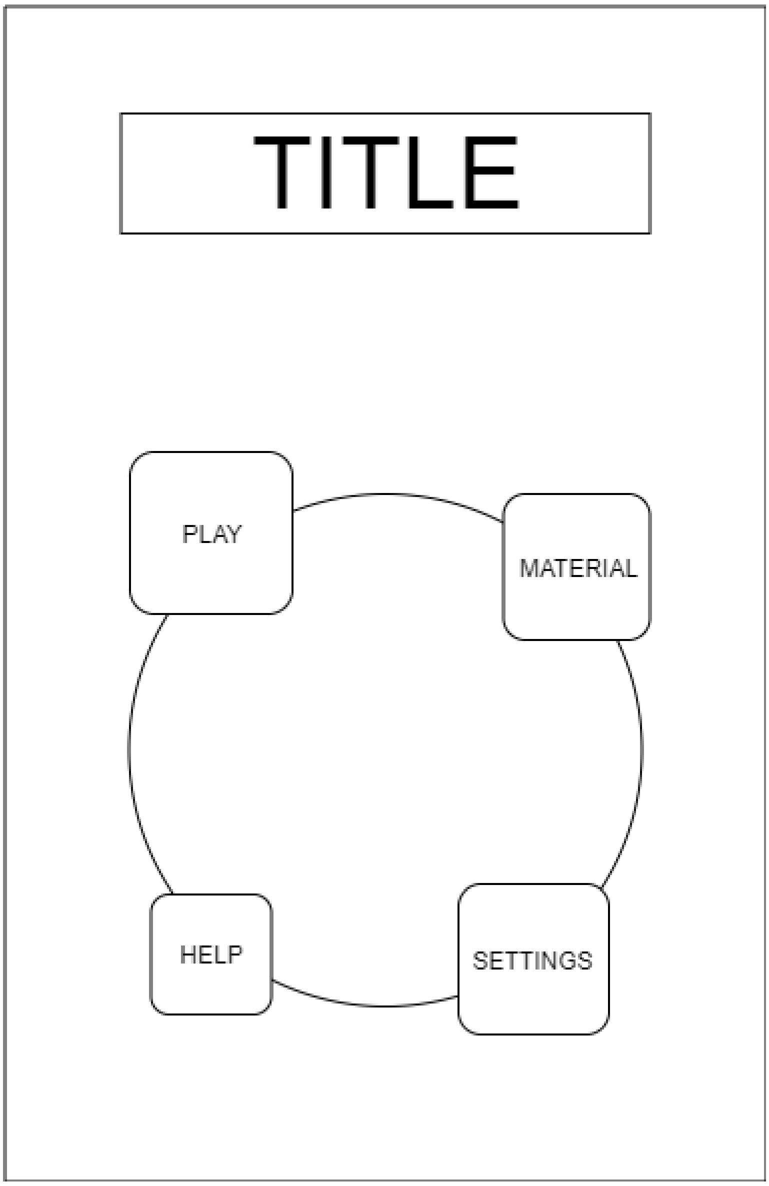

Figure 3. The structure of the main menu.

The Game scene has a heads up display (HUD), which allows the user to switch to different scenes at the touch of a button, to get help with the application, and to switch between what objects to view. Because the application uses augmented reality, the HUD must be very minimal and out of the way of the object being presented to the user. The buttons must be placed in such a way that they are not accidentally pressed by the user and at the same time they must always be pressable. Placing the right button in the right place is also important. Most people are right-handed, so the most commonly used and most important buttons should be placed near the position of the users right thumb. To the left are the buttons created for navigation and other less important functionalities.

This way the interactable viewed content has sufficient space in the middle of the screen. UI design must also take into account the varying sizes and resolutions of device displays. While Unity does pretty good job of adjusting the UI to varying resolutions, the elements on the screen need to be given enough space around them so they will not clip each other or get partly hidden by the screen. As shown in Figure 4, each button and content block are given its own place on screen

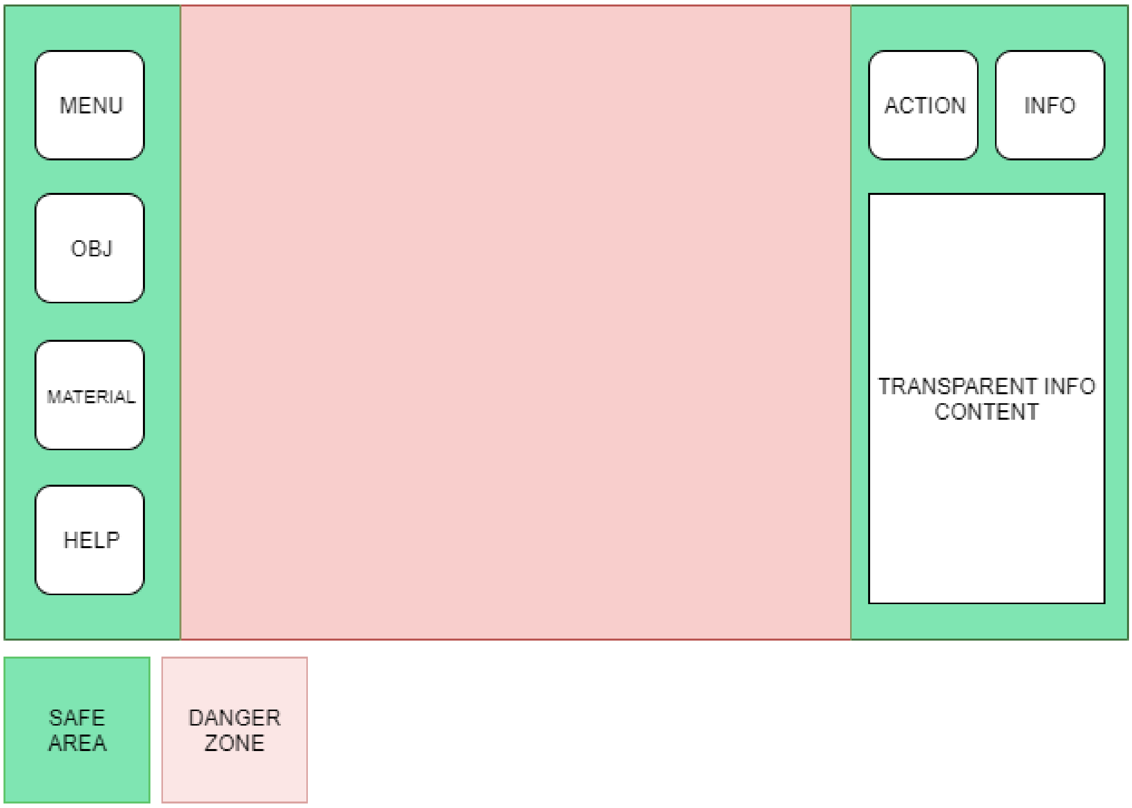

Figure 4. The structure of the Game scene.

In the Material scene, having multiple buttons and options is just a hindrance. For this reason the scene only has a top bar that includes buttons to access main menu and the Material menu. The learning material of the application is viewed in the same way as in almost any other application. It can be dragged, scrolled, and scaled with two fingers as needed. Buttons were placed in the images of the pages, which allow the user to go view the learned subject in the Game scene.

5.3 3D modeling

When the theme is education and medicine, the content of the application must be completely factual and withstand high levels of critique. It has to be perfectly in line with the source material, so the so-called artistic freedom is very limited. Several different sources related to the source material were used for modeling, as these functions and neuroscience elements are described in slightly different ways depending on the source. Images were chosen to aid in creating the 3D models. These images had to be clear, factual, and simple to transform into three dimentional models.

Models to be created included synapse, neuron, and electrical current travelling to an electrode. In addition to these, there had to be modifications and improvements done to the model of human brain and body.

In modeling the performance of different devices was a key factor. Too detailed models, especially when animated, can slow down the application, if not even render it unusable for certain devices. In addition, the very detailed models are large in file size and thus increase the space required by the application on the device. But as this application was for educational purposes, the models could not be low-poly or too simple, there had to be a middle ground. This meant the models had to be designed simple, and the polygon count had to be kept as low as possible while still having smooth models. This needed to be tested during the modelling process.

5.4 Simulations planning and design

At the beginning of the application development, the modeled objects were animated to simulate different neural events. This included f.ex. simulating the nerve impulse going trough a neuron and the different events that happen when the impulse reaches the synapse, like how the different ions and cells act. The simulations were made using Unity's own animation tool, which is easy to use and avoids any problems that might arise when importing animations from a different software. Blender was also viable option for animations, but imports from Blender to Unity can sometimes get tricky or be unpredictable in different ways, possibly even causing corrupted files.

The animations were kept simple in order to achieve a good enough performance while retaining all the necessary information and key events. The ultimate purpose of the application was to help the user internalize what they had read before from the learning material, and if the simulations created were not clear enough, they could be detrimental to the learning process. This included keeping the flashiness at minimum, using simple colors to separate objects from each other and trying to avoid particle effects. Particle effects were to be used only if completely necessary as they can severely impact the performance.

5.5 Gamification

This applications relies heavily on gamification. Originally the application was intended to have actual gameplay in it, but the scope of the application was too big for it to be developed inside the timeframe. Instead of making gameplay, the application is gamified with the use of AR. Gamification is achieved with game-like elements in the Game scene, by visualizing the learning material. The user can interact with objects, resize and rotate them. They can play the animations that simulate different events and watch them happen as close and from whatever angle they want to. This allows the users to get better understanding of the events and help them learn. In addition to helping them learn, interacting with objects floating in the air is fun. The user can even get creative with how and where they place the virtual objects. Achievements and other gamifying elements can be added in futher development.

5.6 Planned testing

Throughout the development of the application, there was continuous testing to resolve any problems immediately while avoiding smaller issues from developing into bigger problems. The testing was performed internally on Unity and on two Android smartphones: Oneplus 3 and Oneplus 6T. These mobile devices well represent the two most important features of mobile devices considering this application: the screen size and the technological age of the device. Both have been tested to be compatible with Vuforia and Unity. Unfortunately these devices were not compatible with Vuforia’s ground plane detection even when the documentation said they would be.

In addition to continuous testing, more extensive testing was carried out during the development stages. A small contribution was also made by client, who was regularly consulted on the status of the application. In addition to testing the functionalities of the application, the reliability of the content was verified by the client in every status check.

This thesis did not include more extensive or comprehensive testing. At the end of the development work, the application was delivered to the client, through which it comes to alpha-testing for an unspecified number of people. More comprehensive testing is part of the potential further development of the application. Testing would have to be done by both engineers and students studying the subjects of this application.

6 IMPLEMENTATION AND RESULTS

The application development started after sufficient background work and a comprehensive plan. Initially, an application body was created, which included the necessary scenes (Game scene, menus and Material scenes), and their interconnection. The necessary add-ons and tools, such as Vuforia, were installed in the project. In addition, at the beginning of the development, Unity's own tools were used to create a framework for the UI of each scene. In addition, the Game scene was prepared for AR functions by adding Vuforia's game elements, such as the AR camera and stage to which the 3D models are projected. The stage is an image set in the Vuforia’s settings that works as the target image when using the application. When the application camera is focused on this image, game objects will appear depending on how they are oriented and laid out in Unity. In the final phase of the application development, a QR code was generated which acts as both a target image and a link to download the application. An empty game object was added to each scene to place the code files required for the scene to function.

After the construction of the application framework, simultaneous design of user interface graphical elements and 3D models was started. When developing an application, it’s smart to utilize previously created assets, our own or those available on the Internet. Unity has its own store where developers can buy, sell, and install content created by other developers, such as game elements and add-ons. For this project, only two good enough free license 3D models were found: the brain and a simple human neural structure. The user interface graphical elements were mostly new designs created by me, as well as other 3D models included in the application, such as the synapse and the neuron.

Once the graphical elements and the core content of the application were completed, it was moved on to integrate the content into the Unity scenes. The most important part was to start with a Game scene where the modeled objects were placed. First, it was important to test the functionality of the objects in practice, that is, to check the scalability of the models and to ensure their integrity. Sometimes the imported models suddenly show broken mesh and clipping surfaces, which might not have been visible while creating the model inside Blender or other modeling software of choice.The objects were then animated and corresponding info texts were added, naming the parts of the objects like the frontal lobe of the brain. The info texts can be toggled on and off with the info button. Learning materials provided by the client were added to their own scenes as a browsable series of images.

After adding key content, the interface was enhanced. The designed graphic elements were placed in their respective positions according to the created frame and tested to make them work. Required pop-ups for object browsing, instructions, and settings. As the user interface was completed, the theme was consistent throughout the application. Partially at the same time as the interface was completed, the operation of the objects to be simulated was programmed. At the end of the development, an official adaptive icon was created to represent the application. The app was built on the Android platform.

6.1 Programming

The Programming was done using the C# language and Unity's own libraries. Unity's own graphical user interface greatly reduced the amount of coding needed. Coding was done in Microsoft Visual Studio, an IDE widely used by professionals for all kinds of development needs. Its quite a heavy to use but it integrates nicely with Unity. Visual Studio features IntelliSense and code refactoring, integrated debugger, and other useful built-in tools.

Programming the skeleton of an Unity application always involves linking scenes, as well as coding the functions of key buttons. Blank game objects were added to each scene to which the written script files were added. Switching between scenes was accompanied by a spinner, which tells the user that the new scene is loading in the background. This is important information for the user who may otherwise think that the application has crashed if the load time gets long. In particular, scenes with large files may take several seconds to load in, even with high-end devices.

Since the application was developed for mobile devices, it was important to consider the orientation of the device in each scene. Each scene includes a script which checks that the device being used is a mobile device and allows the scene to be viewed either vertically or horizontally. The application’s UI was designed in a way that required the Game scene to be viewed in landscape mode, and every other scene in portrait mode. That is because in Game scene the user has to use two fingers when rotating and scaling the objects, which is hard if the user has the device in portrait mode. When it comes to regular usage of mobile devices, the portrait mode is preferred. With tablet devices it's a little bit different, since they are mostly used in landscape mode because of their size. This application was mainly developed for smartphones.

In the Material scenes were only two buttons, both of which led to different menus. For browsing, pages were placed in Unity's own UI objects that allow browsing. These objects also contain code that allow page scaling, and keep the users browsing and scaling within given limits. This allows the user to view images and text more closely while reading the material, but does not accidentally zoom too far or close for the material to go out of sight. Buttons were placed on the images of the materials, and when pressed, the user goes to view the object in question in the Game scene. The name of the image in question is stored in a variable which, when moving to the next view, indicates which object is activated for viewing.

When entering the Game scene, it is checked whether the user has come from the main menu or from a Material scene. The main class checks whether the previously mentioned variable has a value and activates the corresponding object. Otherwise, the 3D model of the brain is activated by default.

All the 3D models have some kind of animation. The user can activate the animation by pressing the Action button, whereupon the code activates that object's Animator and animations, and changes the Action button sprite to one that indicates it being activated. If animation is already activated, pressing the button stops the animation.

The user can activate the Info button at any time to display information about the currently displayed object. A script attached to that button goes trough the array of text objects linked to the active object, activates them, and swaps the button's sprite.

Each 3D model has a script that allows scaling and rotating of the object. Limits can be set for scaling so that the user cannot accidentally make the object too large or too small. In addition, various other related settings can be specified, such as how many fingers on the screen are required to activate these events.

When changing the object being viewed, the main method first deactivates all ongoing animations and then all the 3D models. It also deactivates all the info texts that might have been active. The selected object is then activated and its name is set in "activeObject" variable as a string value. That variable is used everywhere in the script to see which 3D model is currently active. To finish off, the Animator of the select object is set as the current Animator instance

6.2 Graphic design

The design and creation of the graphical elements of the user interface was based on the information learned earlier in education on color theory, best practices, and the latest trends. Creativity is highly valued in graphic design. Knowledge of color theory and graphic design can be utilized in many areas of application development as it’s a large part of user-oriented design.

Graphic design starts with a plan that defines the color palette, overall appearance, and theme. Based on the design plan, a monochromatic blue color scheme was chosen to fit the overall theme of the application. Today's trends of round shapes and soft colors guided the first drafts of the UI sprites and icons. The icons and sprites went trough several iterations trough the development process. All iterations contained similar elements, but they differed for example by shape and outlines, seen in Figure 5.

Figure 5. Drafts of the icons and sprites.

During the design process, different icon sets were tested for compatibility with the application. The final choices were influenced by the distinctiveness/contrast, simplicity, readability, and compatibility of the icons with the AR view. The buttons had to stand out enough from the ever-changing background, but still disturb the viewer's view as little as possible. The user's attention must be on the gameobjects that they are interacting with.

The final decision was to select simple round/rounded icons that were white with black drop shadows. They followed common practices that the users of Android devices are used to. They have a clean modern look but also bring in a little bit of playfulness. Drop shadows are there mainly to also make the buttons stand out from white backgrounds, which is seen in Figure 6. White is a great contrast on almost every background color and it’s rarely the background color itself.

Figure 6. The final icon set.

The main menu is one of the most important parts of the application. In addition to its functionalities, its purpose is to communicate the mood and theme of the app to the user. The main menu shows the application name and various functions such as access to the help section, materials, "game" and settings. These buttons were placed in their own sections on a designed sprite that was inspired by the synaptic structure of the brain, as illustrated in Figure 7. In the center of this element was placed a translucent 3D model of a brain, which rotates about its own axis as the user tilts the device sideways. We can get the data of how fast and far the user tilts the device by using a script that pulls data from the accelerometer of the device This provided a fun and interactive element for the user to watch as the app opened. It also perfectly communicates the theme of the application and what the user can expect from it.

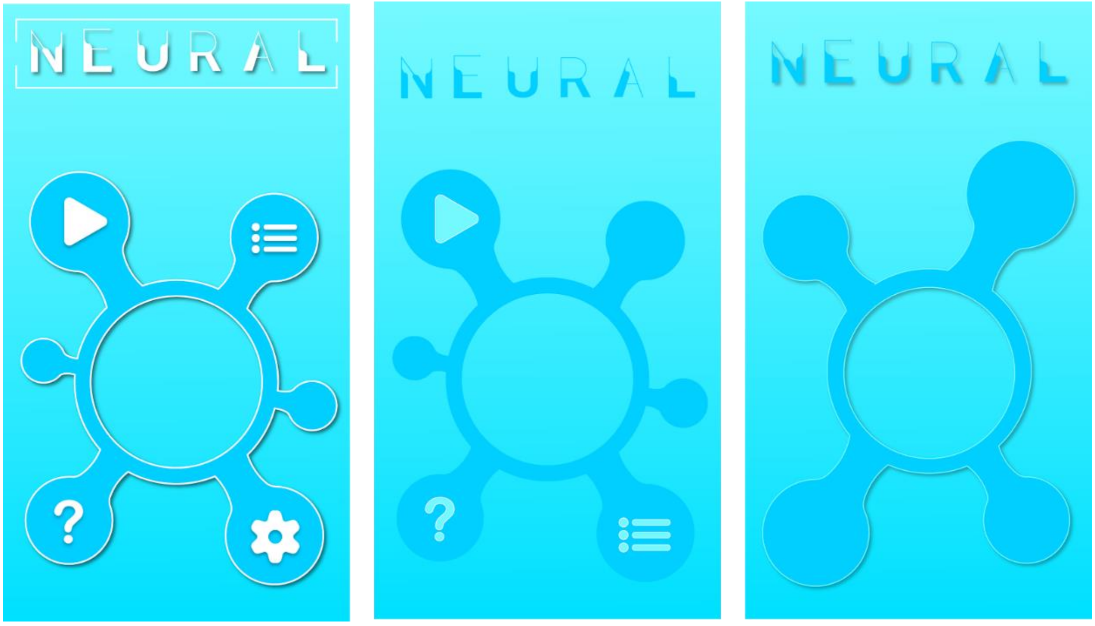

Figure 7. UI drafts for the main menu.

In the Game scene, in addition to the buttons on the HUD, icons were designed for each 3D model. Again, efforts were made to find possible ready-to-use icons on the Internet. Three vector images were found with free license: human head silhouette, brain and human silhouette. These three were utilized in creating the final icons (Figure 8) and the rest were designed and executed by me from the start. The round shapes and ongoing theme were still adhered to. This time rounded squares with drop shadows were chosen as the background for the icons. This rounded rectangular element was used as a backdrop in almost every scene. In general, it served as a background element for pop-ups, but it was also used as a sprite for the buttons in the material menu.Because of the use of Illustrator and vector graphics, this sprite was easy to fit into different usage cases. The font used in the application was Lora Bold and LiberationSans SDF Material.

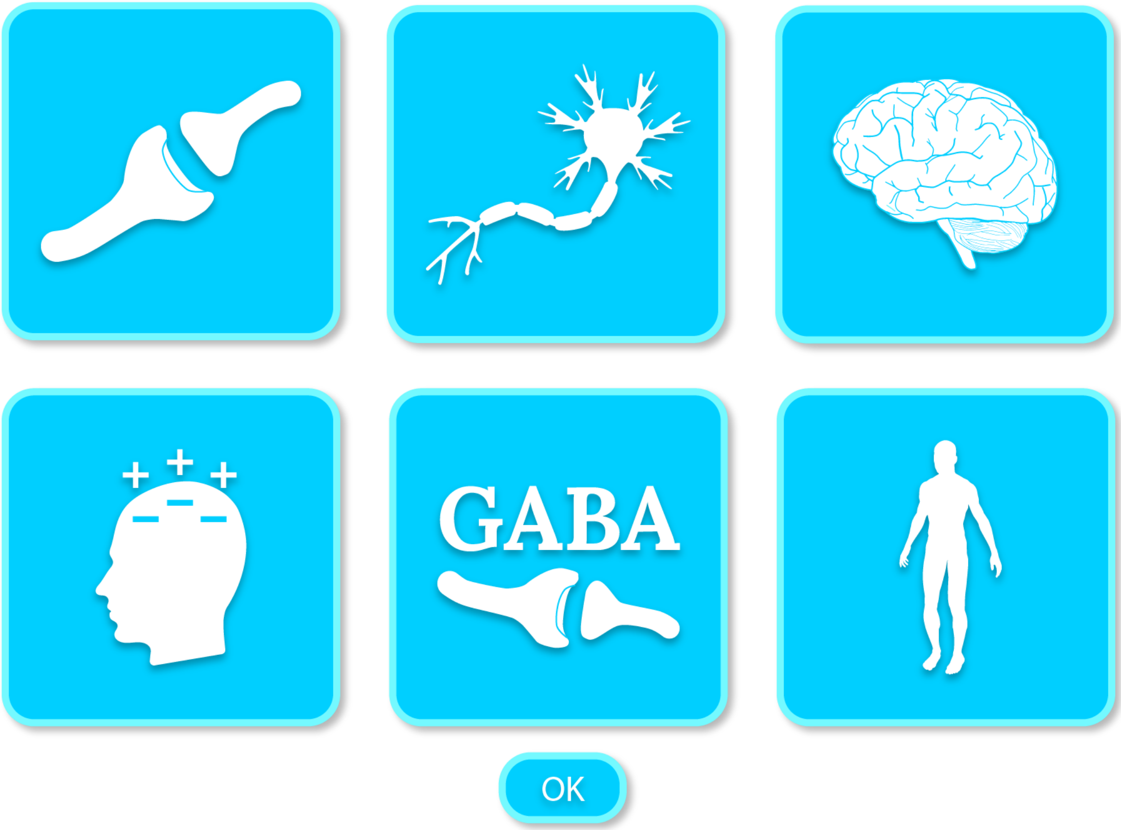

Figure 8. Final icon set for the 3D models.

6.3 3D modeling

When modeling real world objects, especially for teaching purposes, there is a need to be really accurate with the reference materials. The models do not need to be very detailed, but the done work must be factual. Thus, the material acquired for inspiration/examples to follow must also be selected with strong source criticism. The ability to be particularly source-critical and at the same time to find good enough comparators suitable for modeling emerges. As with programming, the ability to effectively search right information and continuous updating of skills are important in modeling. Support for learning is provided by several forums and learning portals, as well as numerous tutorials. The author had previous experience with modeling from school projects and personal work. It included modeling everything from low-poly game elements, to creating realistic models of real world objects.

Challenges with the modeling arose in particular when creating and combining round shapes. Normally when modeling the artists initially create so-called. low-poly models, meaning simple low polygon count mesh. This makes it easier to customize the model as needed and maintain good performance. Only in the final stages of modeling can the model be subjected to different transformation algorithms called modifiers (Figure 9). Blender has a large number of tools that can be used to automate modeling with different algorithms. They range from mirroring to subdivision surface modifier that basically rounds the selected mesh. With subdivision surface modifier we can very easily create round but smooth meshes without having to manually create new vertices and modify them to get the round shape. By creating new edges we can choose how rounded or sharp to make certain parts of the mesh. Sometimes, however, these algorithms create holes and other imperfections in the mesh, requiring them to be manually fixed. That is why we have the low-poly mesh that is easy to alter and fix after trying out different modifiers.

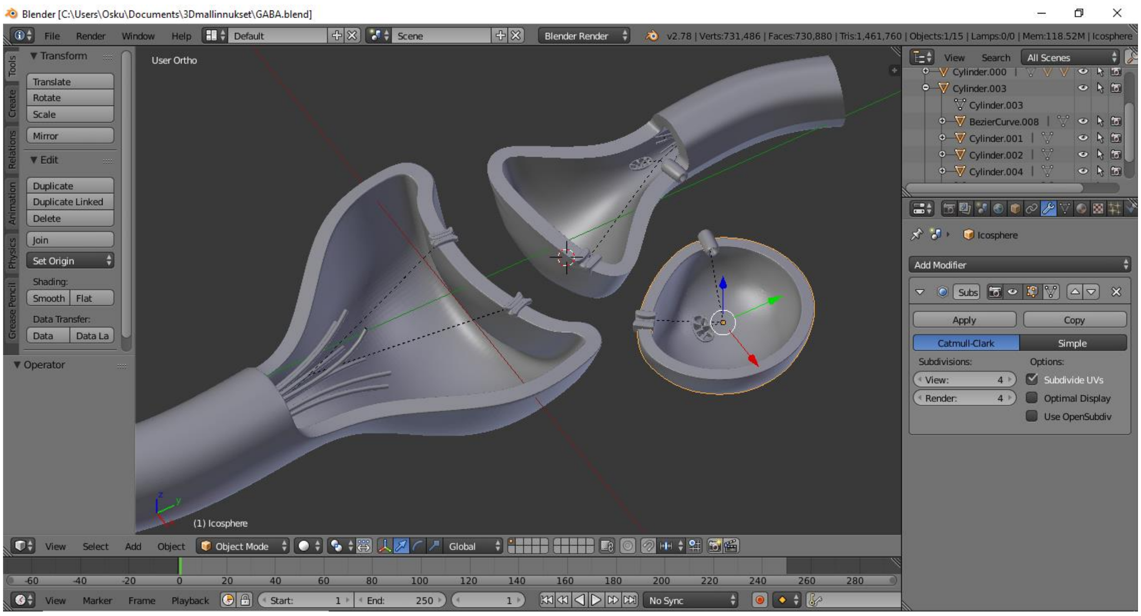

Figure 9. Modeling the synapse with Blender.

Modeling also involves creating materials for objects. These textures can be created directly inside Blender by first unwrapping the object's mesh into a flat 2D image, called UV unwrapping. This can be done manually, but fortunately Blender handles this pretty well automatically by press of a button. When we have the UV map we can for example draw or paste images on it and edit its appearance to create a texture and material (Figure 10). These can then be exported with the model and imported into Unity. Importing Blender created materials require a little bit more work but we can export the textures and maps, and then create materials from them in Unity.

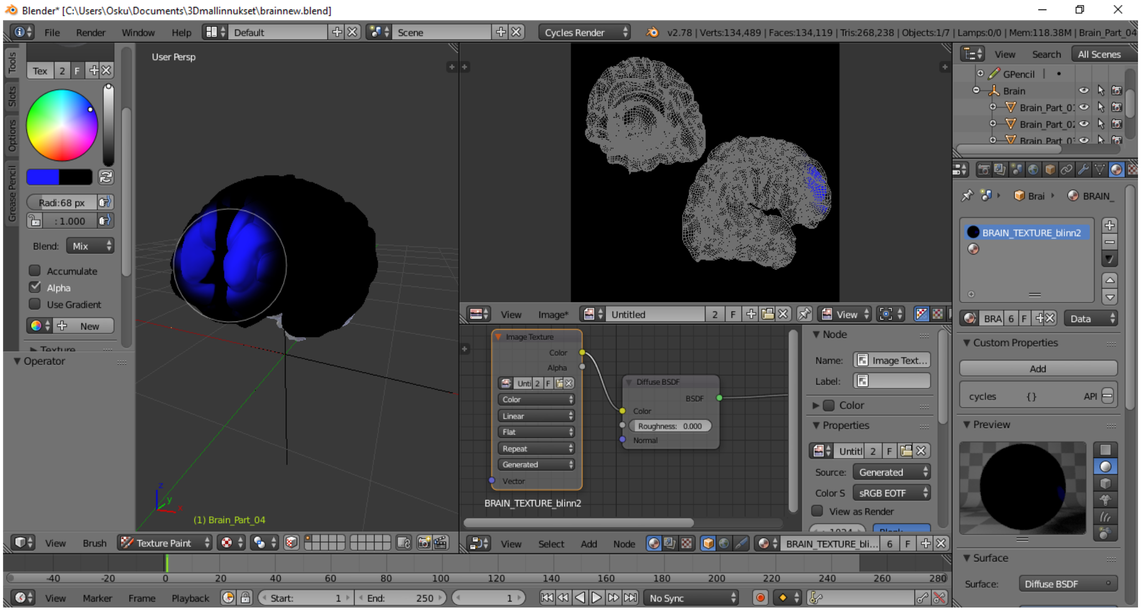

Figure 10. Creating a simple texture and material with Blender by painting.

6.4 Animating

3D models were animated using Unity's built-in animation tools. With each game object is associated an Animator component that controls the animations for that object. The Animator contains animations as well as their work logic. For this application, there was no need for more than one animation per gameobject.

The authors experience with animating at the beginning of the project was very limited. It consisted of a few simple animations related to previous game projects. Fortunately, the Unity’s animation tool is easy to use and the required animations were not particularly difficult but really tedious to create. Animations were created by modifying the position, size, and orientation of the various parts of the gameobjects in the timeline seen in Figure 11. The animations also utilized Unity's particle system tool to automatically create small simulations and effects. With particle systems it was possible to efficiently create multiple floating groups of ions around and in the synapse models.

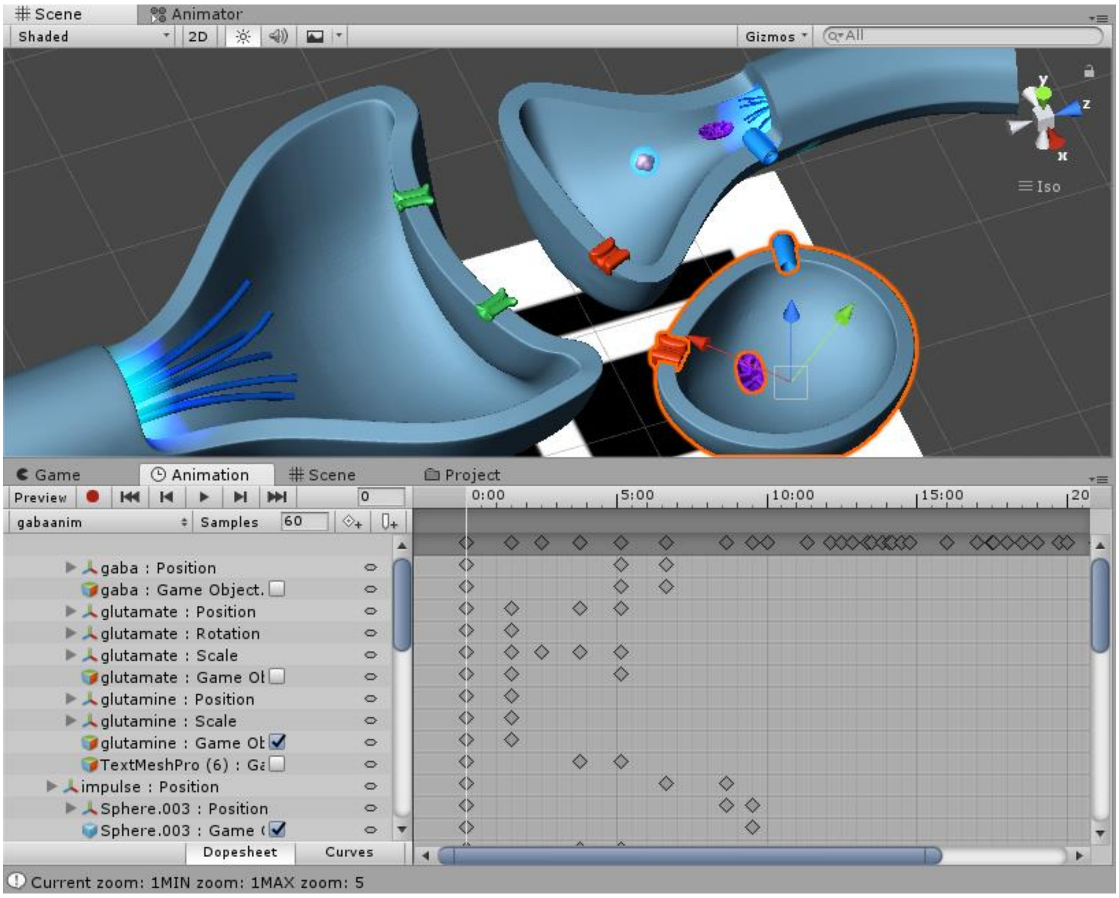

Figure 11. Creating animations with the Unity's integrated animation tool.

Once the animations have been created, their functionality and logic in the application can be programmed using the Animator (Figure 12). The animator is a visual Unity tool that connects different animations together under defined conditions. Conditions can be set for transitions in the code and then adjust settings from the graphical user interface. In this case, a "true / false" condition was set for all animations, where the value of the variable "animated" is set in the code by pressing a button. This triggers an animation that starts and then stops when the button is pressed again and the value of the variable is "false".

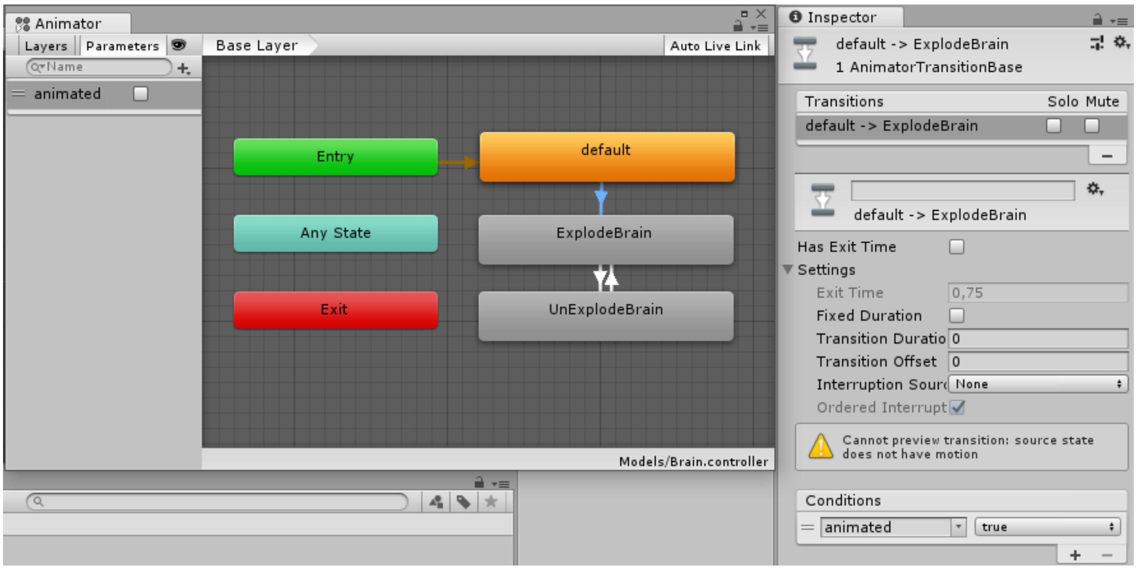

Figure 12. Unity's Animator tool.

6.5 The end result

There were some problems during development, but these were often minor or due to incompetence. They were often corrected after a little debugging or searching the Internet for more information. The biggest problem encountered was the incompatibility of the test equipment with Vuforia's Ground Plane functionality. During the design process, it was intended to explore the potential of this functionality and its compatibility with this project. However, this was only considered as a good addition instead of a required functionality, and previous testing had already shown it working suboptimally.

The application development ended with a finished product containing the elements requested by the client and perhaps a little more. The application was tested for functionality and provided to the client for his own testing and use. Here is a demo of the final application and a QR code where it can be downloaded, screenshots presented in Figures 13 trough 16.

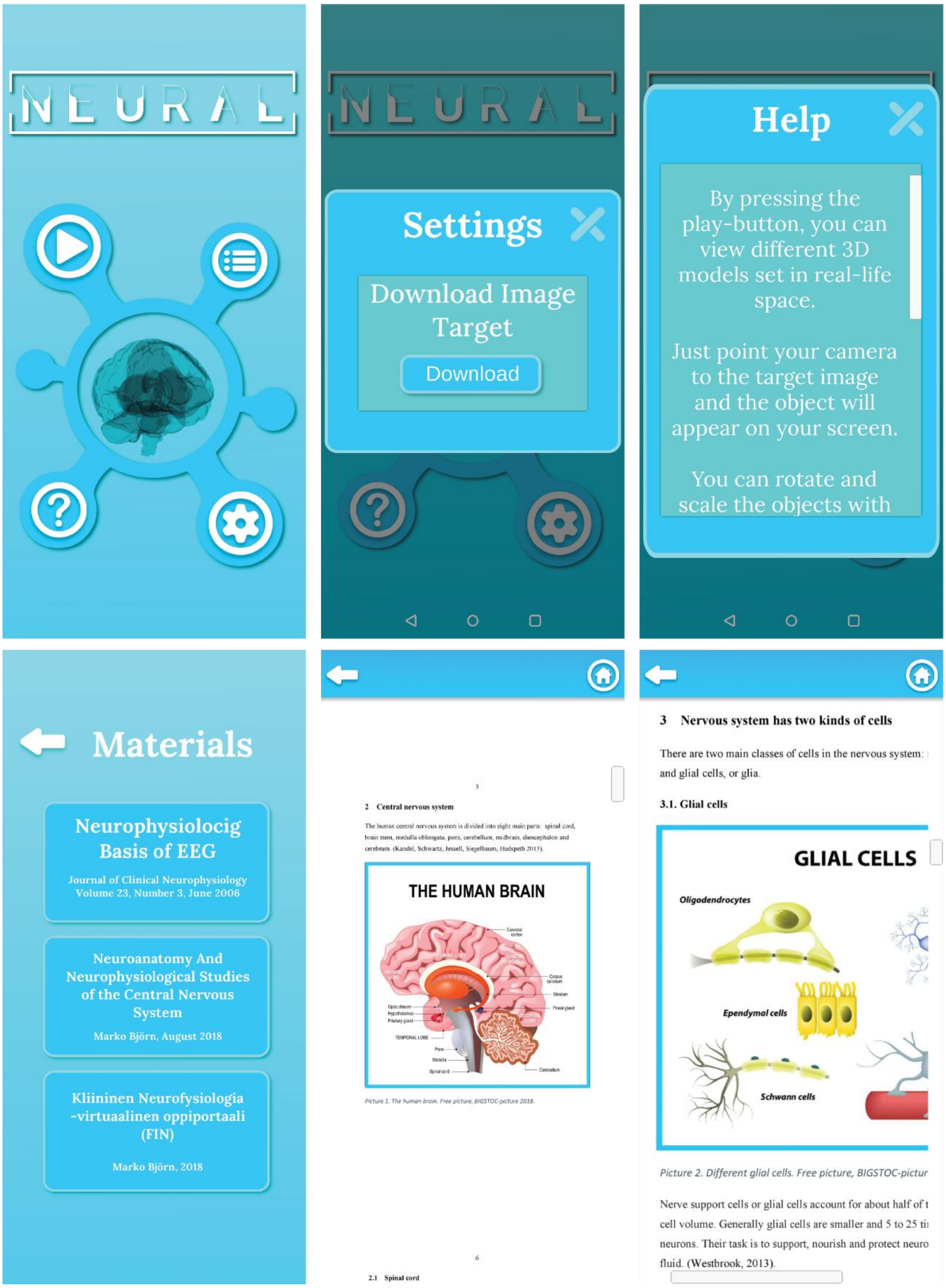

Figure 13. Final application part 1.

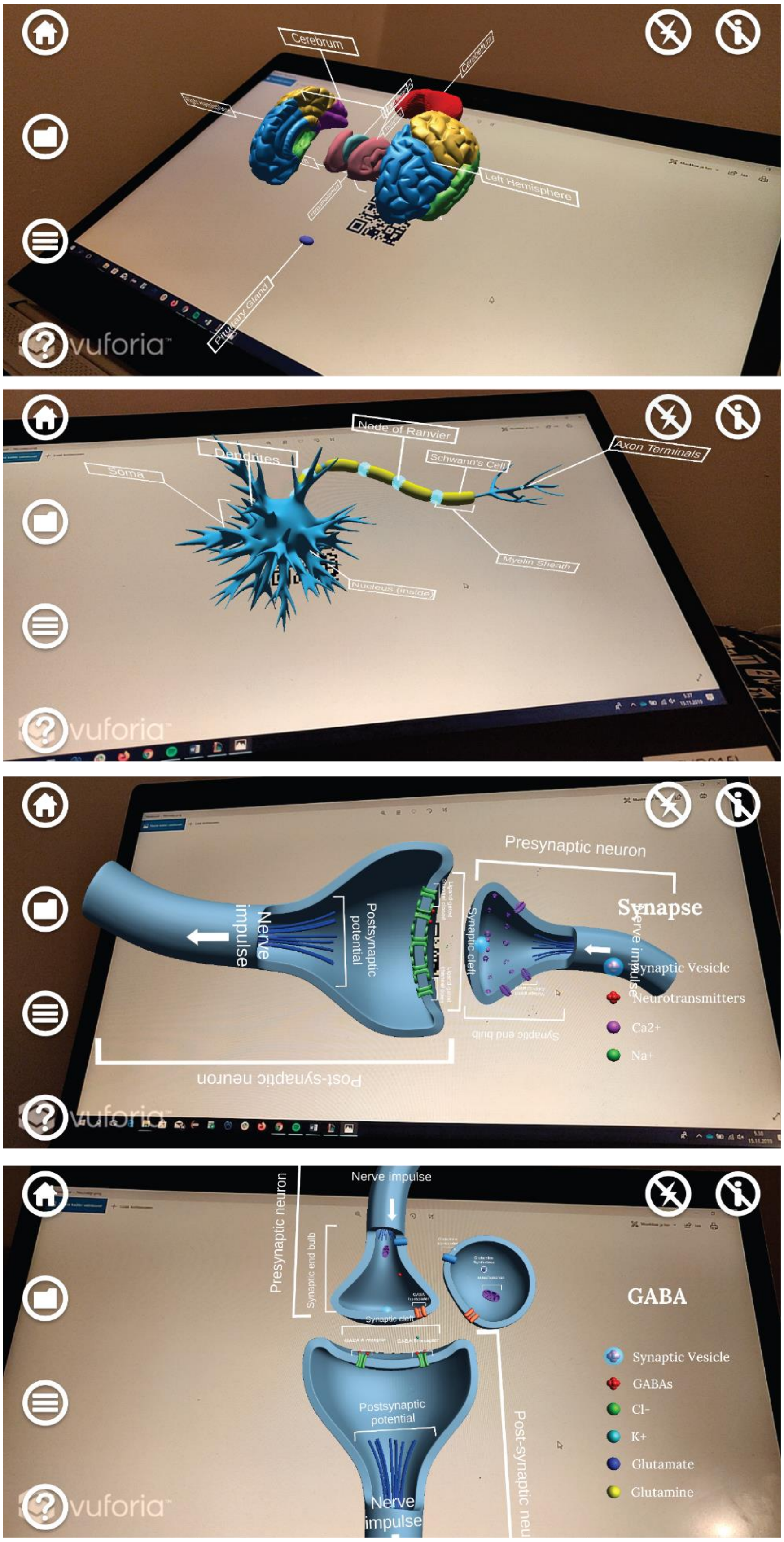

Figure 14. Final application part 2.

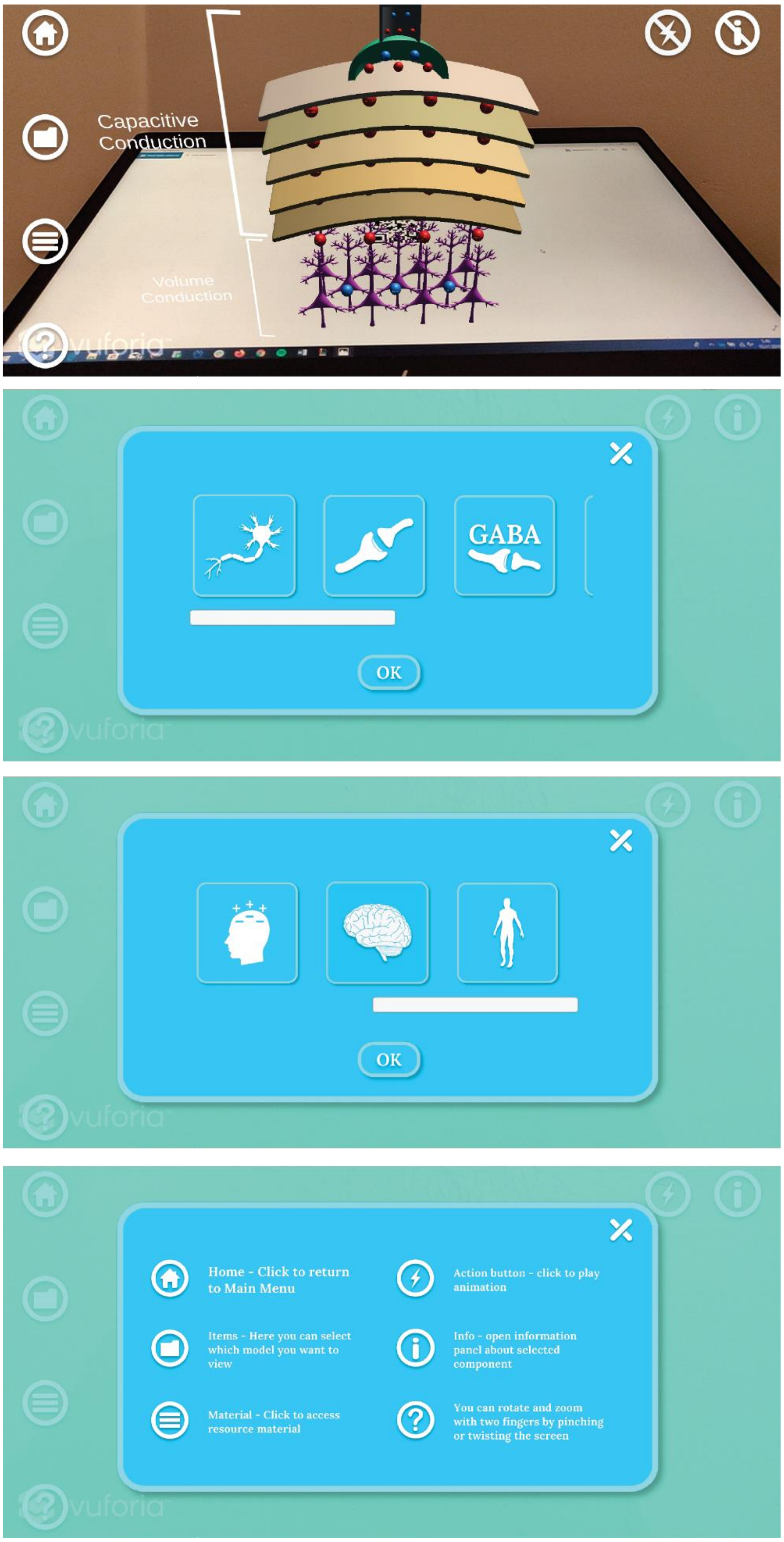

Figure 15. Final application part 3.

Figure 16. QR code to download the application and to use as target image.

7 USE CASE

A typical use case for this application would be done by a student studying the field outside of class. The application can also be used for teaching aid in class when covering a subject that is handled in the application.

First the user, a student, would want to study the biological functions behind the workings of EEG. The student would then start the application and arrive at the main menu. In the main menu the user has an option to go straight to the Game scene or to first take a look at the theory of studied subject. Most of the time the user would click the Material menu button and get few options to read up on some theory.

The user selects a material based on the current subject that they are studying in class. They will read some theory and go trough the material until they arrive at a picture of, for example, a synapse. They have now read up on the way neurons work and what happens in a synapse, so they can click on the picture of the synapse and be taken into the Game scene.

In the Game scene the model of a synapse is active but the user will not see it yet. Now the user has to point their device’s camera to the QR code they downloaded and printed themselves, or that is provided by the teacher of the class. The synapse will now appear on the screen, floating in the air above the QR code the camera is pointed at. The user can now make the model bigger or smaller, and rotate it. The user will study the synapse in 3D, then click on the info button to activate all the texts describing different parts of the synapse.

After studying all the parts of the synapse, the user will click on the Action button. This will activate the animation for the synapse. This animation simulates the basic function of the synapse, now visualizing the theory the user just read up on in the Material scene.

After learning enough of the synapse, the user can now:

1. select a new object to view
2. go back to the reading material
3. exit to the main menu.

8 REFLECTION

The purpose of this thesis was to gamify the learning and teaching of neurosciences. The goal was to create a functional Android application that can display 3D objects in AR, as well as have the user interact with those objects and learn about them. As time went by, the systematics suffered and the Unity project and some code snippets were left a little messy. The design plan created in the beginning of the development process was good and it was followed troughout the development as well as possible. Along the way, parts of the application were left out by the decision of the developer and the client. This included simulating an epileptic event, which could not be done because the science behind it is not completely known. This is in line with the importance of providing greliable information and keeping everythin gin the application completely factual. Reason for few other minor functionalities being left out, was the lack of time. Although the implementation was little different from the original plan, the application stayed true to most of the plan.

Further development could include the addition of neuroscientific materials to the application and the creation of more 3D models. Animations could be improved, and new ones created that more precisely and closely simulate certain events. It would be useful to add an alternative setting to mirror the buttons in the Game scene for left-handed users. It was also intended to add Ground Plane support to the application, the ability to project objects onto any flat surface such as a table or floor. However, this was left for further development, as the devices used did not support this Vuforia function, although the documentation said it should. This feature would be very useful as the user would be able to use the application anytime and anywhere, without any dependencies. To help the user more with using the Game scene, there could be added animated instructions on how to operate it when the user first enters the scene.

The feedback from the client and the supervisor of the thesis was good. At regular intervals during application development, we discussed the content of the application and the various needs, as required for the design priciples used. The client is extremely pleased and eager to use the new AR technology in particular as a teaching method and is impressed with the way various models can be viewed and handled. The client would have liked the ground plane option as it would have made it easier to use and share the application. In addition, during the development, there was interest in this application from outside parties.

The process of developing this application offered great amount of learning opportunities in both software development, and project management. It showed how important and useful both user-centered design and participatory design are. These design priciples offer greater understanding of the content of the application, and the way the actual user views the application. It also highlighted the importance of good communication between a developer and the client. The design process has to be inclusive and realistic so the project will stay on schedule and there are no surprise costs or problems. This will also generally lower the chance of other conflicts and stress. The design process can be considered the most important part of software development, as it dictates the whole structure and ways everything is done.

In conclusion, the goal was reached with minor setbacks during the development. The workflow suffered towards the end of development due to adding new content not planned originally. This pushed back the deadline and finishing of the application. Everything that was done, was done well, but there is always room for improvement. There is a multitude of functionalities and content that can be added in further development, like achievements and previously mentioned new models.

REFERENCES

99designs. How do you choose colors for a healthcare logo? https://99designs.com/logo-design/psychology-of-color/healthcare. Referenced 10.11.2019.

Augmented Reality. https://www.merriam-webster.com/dictionary/augmented reality.

Babich, N. (2020). 7 Basic Rules for Button Design. UX Planet. https://uxplanet.org/7-basic-rules-for-button-design-63dcdf5676b4. Referenced 9.2.2020.

Clement, J. (3.12.2019). Instagram: active users 2018. https://www.statista.com/statistics/253577/number-of-monthly-active-instagram-users/. Referenced 4.12.2019.

Clement, J. (23.10.2019). Snapchat daily active users 2019. https://www.statista.com/statistics/545967/snapchat-app-dau/. Referenced 15.11.2019.

Elizarova, O., Briselli, J. and Dowd, K. (2020). Participatory Design in Practice | UX Magazine. UX Magazine. https://uxmag.com/articles/participatory-design-in-practice. Referenced 13.02.2020.

Casali, E. (2020). Co-design and participatory design: a solid process primer. Intense Minimalism. Available at: https://intenseminimalism.com/2013/co-design-and-participatory-design-a-solid-process-primer/. Referenced 12.02.2020.

Fabian, A. (5.11.2019). UI Trends 2020: Top 10 Trends Users Will Love. https://uxstudioteam.com/ux-blog/ui-trends-2019/. Referenced 10.11.2019.

Growth Engineering. (2020). What is the Definition of Gamification and What Does it Mean?. https://www.growthengineering.co.uk/definition-of-gamification/. Referenced 13.02.2020.

Interaction Design Foundation. What is User Experience (UX) Design? https://www.interaction-design.org/literature/topics/ux-design. Referenced 10.10.2019.

Karhu, J.(2013). Lisätty Todellisuus, 5–15. Jyväskylän Ammattikorkeakoulu. http://urn.fi/URN:NBN:fi:amk-2013121821760

Mayfield Brain & Spine. EEG Electroencephalogram: Mayfield Brain & Spine. https://mayfieldclinic.com/pe-eeg.htm. Referenced 25.01.2020.

Moss, A. (9.4.2019). 20 Augmented Reality Stats To Keep You Sharp In 2019. https://techjury.net/augmented-reality/#gref. Referenced 13.11.2019.

Olejniczak, P. (2006). Neurophysiologic Basis of EEG. Journal of Clinical Neurophysiology, 23(3), 186–189. doi: 10.1097/01.wnp.0000220079.61973.6c

Partners - Vuforia. https://unity3d.com/partners/vuforia. Referenced 25.10.2019.

Pragmatic Institute, J. M. Software Development Pitfalls: Planning. https://www.pragmaticinstitute.com/resources/articles/software-development-pitfalls-planning. Referenced 13.11.2019.

HUSLAB, S. V. Elektroenkefalografia (EEG), kuvantamiseen yhdistettynä. https://huslab.fi/ohjekirja/4912.html. Referenced 10.11.2019.

Vox. (26.1.2018). Why danger symbols can't last forever. https://www.youtube.com/watch?v=lOEqzt36JEM. Referenced 10.9.2019.

Wheeler, A. (21.1.2019). Bad News for Augmented Reality as Multiple Startups Fold. https://www.engineering.com/ARVR/ArticleID/18257/Bad-News-for-Augmented-Reality-as-Multiple-Startups-Fold.aspx. Viitattu 13.11.2019.

Zandan, N. (2020). Neuroscience and Visual Aid Effectiveness. Quantifiedcommunications.com. https://www.quantifiedcommunications.com/blog/neuroscience-and-visual-aid-effectiveness/. Referenced 12.02.2020.