A report submitted in partial fulfilment of the regulations governing the award of the Degree of BSc. (Honours) Computer Science with Games Development at the University of Northumbria at Newcastle

Project Report

Developing an Immersive Educational Tool for Training Nurses and Reducing Misdiagnosis by Creating Engaging Learning Experiences

Andrew Nattress - W17009550

2020/21

Word Count: 11,762

Supervisor - Alan Godfrey
Secondary Marker - Hua Mao

Software Engineering Project

Declarations

I declare the following:

(1) that the material contained in this dissertation is the end result of my own work and that due acknowledgement has been given in the bibliography and references to **ALL** sources be they printed, electronic or personal.

(2) the Word Count of this Dissertation is 11,762

(3) that unless this dissertation has been confirmed as confidential, I agree to an entire electronic copy or sections of the dissertation to being placed on the eLearning Portal (Blackboard), if deemed appropriate, to allow future students the opportunity to see examples of past dissertations. I understand that if displayed on eLearning Portal it would be made available for no longer than five years and that students would be able to print off copies or download.

(4) I agree to my dissertation being submitted to a plagiarism detection service, where it will be stored in a database and compared against work submitted from this or any other School or from other institutions using the service.

In the event of the service detecting a high degree of similarity between content within the service this will be reported back to my supervisor and second marker, who may decide to undertake further investigation that may ultimately lead to disciplinary actions, should instances of plagiarism be detected.

(5) I have read the Northumbria University/Engineering and Environment Policy Statement on Ethics in Research and Consultancy and I confirm that ethical issues have been considered, evaluated and appropriately addressed in this research.

SIGNED: anattress Andrew Nattress 11/05/2021

Acknowledgements

I would like to thank my dissertation supervisor Alan Godfrey for his support and guidance throughout the project.

I would like to express my appreciation to my anonymous tester for their expert advice and feedback with regards to the product created.

Additionally I would like to thank my employer, Boxmodel Digital, and specifically Alan Easton for the opportunity to develop a project I am passionate about and the education I received while working with them for the past 2 years.

Abstract

This project aims to aid in the training of medical staff using immersive virtual reality software to create a better learning experience. The virtual reality education engine I created primarily targets the medical field due to a problem identified in the initial project proposal and developed during the Terms of Reference. There are a high number of cases of medical misdiagnosis in the United Kingdom, which is a huge problem as it costs the NHS a lot of money and in many cases severely impacts people's lives. During my research it became clear that this problem could be resolved with better education that engages learners and creates a memorable experience. During the Analysis I found significant evidence supporting my theory that virtual reality based learning has huge potential to transform both classroom-based and remote learning.

Early research also discovered that virtual reality headsets have the potential to be too expensive for smaller, lower budget educational institutions, this led to a key requirement in my specification that the system must be adaptable. If the client requests a non-virtual reality based solution it should be possible to build this from my existing solution with as little effort as possible.

The technical complexity of the project largely comes from making it as scalable as possible, the system I developed must be capable of adapting to an augmented reality solution, or adding a simulated patient to the virtual environment that the user can diagnose. Due to the lack of a formalised client for the duration of the project I was not able to derive a specification of their needs, so my employer's need to target as many potential client requirements as possible became the priority for creating a successful specification for the project.

Introduction

Why This Project?

The idea for this project was proposed to Boxmodel Digital, my employer, by a potential client who has lots of experience in the nursing industry and is currently employed by the Cumbria, Northumberland, Tyne and Wear NHS Trust as an educator. He was interested in developing an idea for an augmented reality application that could improve nursing education by providing a more memorable learning experience that has a greater impact on his students.

After lots of research, the project became a virtual reality application as we agreed that it had greater potential to deliver an immersive experience that students will not forget than traditional educational software platforms such as desktop software. This project is an idea that both me and my employer are passionate about, believing it has huge potential for the company to show off its virtual reality capabilities, attract new clients for real time 3D applications and to create a high quality product that we are proud of.

Virtual reality is also an industry that I am particularly interested in, I believe it has the capability to transform many industries, make work safer through the use of better training and immersive simulation and make everyday software engaging, fun and more productive. As the virtual reality industry continues to grow I am aiming to help my company build a strong reputation for delivering quality software that solves problems and leads to clients earning more money. I believe this project has the potential to show that, by creating a solution that impresses the client, but also has a wide array of potential use cases outside of the single client. Most industries need job training, and virtual reality has been shown to have huge potential to enhance e-learning, for example in a case study by Oculus (2020) Hilton reduced their training time from 4 hours to just 20 minutes after switching to virtual reality.

Aims and Objectives

Aim: To develop an educational tool, that can help to reduce cases of misdiagnosis in the NHS.
Objectives: ● Investigate effective methods of teaching, such as Kolb's experiential learning cycle, and improve learning with the use of immersive technologies by implementing effective teaching methods.

Aim:

Develop a requirement specification based upon my literature review, research gathered from my prototype demonstrations and my past interviews with CNTW employees.
Objectives: ● Identify good teaching methods from my literature review and explain how I will implement them in my project report. ● Develop a virtual reality application that meets the requirements of the specification I wrote.

Aim: Learn how to better design software so that it improves the result of the product.
Objectives: ● Study good UX design for virtual reality in my literature review and implement it in the application. While this area is still being researched, I aim to enable the user to pick up and use the application easily. ● Produce chapters of the project report as I complete objectives, this will allow me to explain how and why the decisions I make improve the software design.

Aim: Review the solution I developed by providing users who are training to be nurses access to the application in order to evaluate its effectiveness and receive feedback.
Objectives: ● Test the solution by having nursing students try and review the virtual reality experience. ● Analyse the results from my project testing, discuss how this affected my designs and how the results compare to the findings of my initial research.

Aim: Complete an evaluation of the product and its development process.
Objectives: ● Discuss the success of my implementation, based on the test feedback and the response of the client. ● Evaluate the development and my effectiveness in designing the software and implementing the needs of the client..

Product Overview

The final product I have developed is an educational platform in which student nurses can experience immersive learning and study the symptoms of medical conditions. The product is designed to reduce occasions of misdiagnosis in medical environments which is a huge

problem in the UK, there have been over 4,000 successful compensation claims in three years (NHS Resolution, 2018). The virtual reality application does this by providing detailed views of difficult to diagnose conditions such as Erythema and a good learning environment for studying with important condition information and the means to test your knowledge.

The system includes a method of viewing the conditions on the body, a screen to read information about the conditions and a quiz system that allows users to test themselves on anything they study in the session. The system is built with flexibility in mind, allowing more conditions to be added very easily and for the same condition data to be usable in other applications, for example, if the client decided they would also like an Augmented Reality app, the data and much of the codebase would be reusable in this application.

Approach Summary

I utilised the Unity game engine to develop the virtual reality application, targeting the Oculus Quest as the product's primary platform. My previous experience with Unity helped me to architect the program well, using the design theory "high cohesion, low coupling", to maximise code reuse and readability for any developers that may join the project in its future development.

I also used the Oculus integration SDK as this allowed me to use Oculus specific functions in the Unity engine. A couple of months after starting the project the XR Toolkit by Unity released, which would have been a better fit for the project as it supports many brands of headset as opposed to just the Oculus suite. While this would have been a better fit, the Oculus integration includes animated hands and several other features that would have taken extra time to implement. The application will still be possible to run on all headsets if required by the client, it just requires more effort to switch SDK's based upon headset choice.

In order to achieve the best architecture I could design in Unity I planned the system first using simple UML diagrams. This allowed me to identify and remove dependencies wherever possible ahead of beginning development. I particularly focused on having 0 dependencies between separate groups of features as visible in Appendix 3, none of the "core" classes rely on any classes in other sections such as the "VR" group. This means that when moving to new platforms all of the core features should be reusable, even classes such as "tablet controller" will not be limited to use in the virtual reality app, it will work just as well with a full screen canvas user interface.

This was achievable due to design decisions I made such as the use of a ScriptableObject for Selected Condition, meaning that many scripts can share access to the variable without relying on a singleton. This will allow me in the future to add a compare system between 2 conditions if needed and also means that I can control events whenever it is set using the GameEvent class.

Another big design decision that will benefit future changes is the use of an interface to control spawning for the ConditionManager class. This means that if I have multiple instances of Condition Manager they can have their own spawn function. This will benefit me when I develop for new platforms because augmented reality, as an example, will require the user to decide where to spawn the condition, whereas the virtual reality app attaches it to the skeleton.

Analysis

Critical Review of Literature

Virtual reality is a rapidly growing industry and headsets are finally providing a great standalone experience and becoming more affordable. This has created new opportunities to utilise virtual reality for educational applications. There is significant evidence that immersive education can improve the learning experience of users and increase retention of information as discussed by Huang et al (2019). This chapter will survey the use of educational software, in order to assist the development of a virtual reality educational tool with the purpose of reducing misdiagnosis in the nursing field. Nurse education has been identified as a leading factor in the increasing cases of medical misdiagnosis, and this application is intended to reduce these cases.

Evaluating the Effectiveness of Educational Software

Introduction

Educational software is utilised around the world in schools, universities and even in the workplace. Virtual training software was proven to increase learning speed and satisfaction in a Cambridge University study by J M Frank-Law (2015). A large number of educational applications are used daily (Uziak, 2018).

In order to judge the effectiveness of software as an educator, a method of evaluating it is needed, there are many factors to take into account, such as the student's change in grades, student satisfaction and the intended use case of the software.

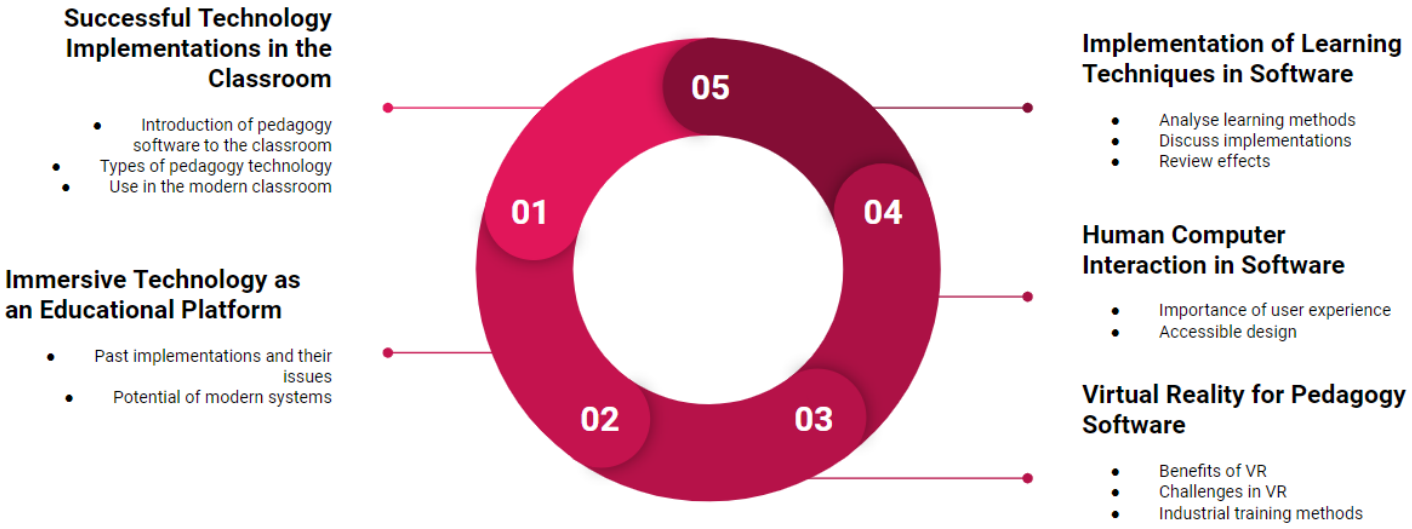


Figure 1. Considerations for the development of modern immersive pedagogy software.

Transition of Technology into the Classroom

Computers can be a useful tool for teaching problem solving and critical thinking skills (Ranasinghe, 2009). Computers and similar technology such as interactive whiteboards have become a classroom staple in the 21st century (Beauchamp, 2004).

In the years since technology has been integrated into the classroom, many types of software have been developed to increase the educational capacity of computers. For example, Blackboard is an online learning portal that is used to distribute course materials, organise and track student progress and to improve communications between students and staff. Students reported many benefits of using the software, including quick feedback, better availability of course content and boosting of organizational and time management skills (Bradford, 2007).

Duolingo is a different kind of educational software, a mobile application that is intended to educate, helping its users learn a new language. It is proven to be effective at teaching and it is greatly appreciated by students who liken its teaching methods to playing games (Vesselinov, 2012).

Virtual Reality in Education

Benefits

Virtual reality is an active experience that forces the user to take part in active learning (Hussein, 2015). Engaging the user in active learning is proven to boost learning progress (Park, 2014) and will create a memorable experience that will serve as good word of mouth for the educational software developed. The benefits of virtual learning are still being proven but many studies including Coppola (2002) show that students of virtual learning perform better in tests than those in instructor-led classes. This proves that well-designed software has the potential to improve learning.

Challenges of Developing a Virtual Reality Application

Several studies have shown that learning can be less effective in virtual classes such as Allison's study "Virtual Reality for Education".

There are several shortcomings that often limit educational virtual reality software leading to students not being as productive as usual. One factor is that the application has to keep the user interested, which can be difficult in long lessons (Allison, 2000). For this reason, targeting short sessions might be beneficial to the user, the article suggests that conveying just the important information without including "filler content" should lead to improved results.

Another major issue that can arise in Virtual Reality is that the controls of applications are often difficult for first-time virtual reality users to understand and remember. This made the learning process slow and challenging for the students, which would be detrimental to an educational platform designed to be used in short sessions. For this reason, user experience design must be a priority for any educational software being developed. Applications designed for repeated use should focus their user experience design on efficient navigation and control of the application, making it as easy as possible for advanced users to use the application. When designing for short periods of time, however, efficient navigation should take a back seat to easy to use and intuitive design, the application should lead users to important sections and guide them through the educational experience.

Industrial Virtual Reality Training Software

Industrial virtual reality training software has been a growing industry for several years now, and it has proven to be an effective tool for reducing costs, avoiding exposing learners to dynamic environments, and preserving the integrity of the production chain as discussed by Naranjo (2020). For example, Hilton developed training software for their staff designed to create empathy with their guests, helping staff understand the repercussions of badly handled interactions. They found that the software reduced in-class training time from 4 hours to 20 minutes and that 87% of their team members changed their behaviour after using the application (Oculus For Business, 2020). Similar concepts should be applicable to the medical industry as exposing learners to a realistic hospital environment would prepare them for the real experience. One of the biggest costs for medical education is malpractice insurance, moving more training into virtual environments where harm cannot be done to patients will cause malpractice insurance prices to go down as payouts will be reduced.

Criticism of Literature

While there are lots of studies into education via software, none provide a conclusive answer to proving the best combination of virtual and in-person learning. This is challenging as it is difficult to find a sample size large enough and to have the resources to test any situation. The above studies do however prove that there is a place for virtual reality software in education and that it can, when used in an appropriate manner, significantly benefit learners. Virtual reality software provides opportunities that are not available through other educational mediums and can create a more engaging experience that benefits the education of its users.

Despite the fact that most of the literature analyzed in this section was created to inform the educational design of software, most have very little pedagogical reasoning for their conclusions. A greater focus on the science of learning would have been beneficial to the design of future systems. This could identify whether the information learned was retained over long term lengths and also analyse how the novelty of virtual reality impacts the results. This is important as if virtual reality continues to grow or the users spend lots of time with the application, the novelty factor will wear off over time. For example, Hussein (2015) dismisses novelty being a factor in virtual reality education, however Park (2014) says that memorable experiences encourage learning. The memorability of the experience will diminish as novelty wears off so it should be an important factor when studying effectiveness.

This would all be useful information when designing the application as it would inform decisions as to how the virtual reality application supplements learning and how the system should be used in educational settings. Educational software can act as either a supplement to other forms of learning or as the main information source for learning, while it would be helpful to know this before development begins, designing the software to be useful in either case will be helpful to allow for greater flexibility.

This chapter will also seek to research teaching methods that can be implemented into a virtual reality application in order to create a product that teaches as effectively as possible. These are very important to the application, which is intended to be used to improve the effectiveness of education because current methods are not enough.

Human-Computer Interaction in Virtual Reality

A lot of factors can affect the impact of education on a learner. My prototype-based research suggested that user experience design of the software must be a priority when developing. Further research led to an article by Kraut discussing designing sophisticated user interfaces suggesting that they are vital to the learning process. Reducing user friction makes the learning process easier and students can concentrate on retaining information.

Importance of User Experience

As Kauhanen's (2017) study discovered, audio effects are an essential part of an immersive experience and users are distracted when it is not included. It also found that good audio design could be a central part of immersive experiences, engaging users and delivering feedback. The balance between ambient sounds and the use of event sounds is essential to creating a welcoming immersive space for the user as Serafin writes. Implementing basic ambient sounds can help the user feel comfortable in the virtual environment. Event sounds are important for providing feedback to the user, getting their attention and should be varied to not desensitize users to their effects.

As Ashtari's (2020) study showed, developers usually find that the user experience design guidelines for virtual reality applications provided by platforms such as Apple and Google fail to provide good results for all situations. Ashtari identified that guidelines were particularly lacking for interactive features and that they are aimed towards simple applications. This means that applications that go beyond the guidelines usually develop their own methods of interaction, Kersten (2018) discusses the methods used at their company in order to develop user interaction that was intuitive and easy for users to use.

Accessibility

Good user-interface design in virtual reality is a developing area of expertise (Yong Min Kim, 2019). Chen (2019) discusses the steps to creating an accessible user interface design; identify potential exclusionary interfaces, explore solving strategies of similar software, develop a solution that meets your needs using a persona, review its compatibility with all

past personas and iterate. This process can help create an experience that is inclusive to all people and that is best suited to your application. Accessibility is a design concern in virtual reality and developers must be conscious when designing their platform in order to be accessible to as many people as possible.

A major accessibility concern is for women in virtual reality, an Oculus study showed that lots of factors affected people's comfort in virtual reality (Gokalp, 2019). A primary factor for women was the virtual environment they were in, they found that the majority of women, when placed in a science fiction styled environment, felt uncomfortable and scared in virtual reality, where men generally found it exciting, compared to a homely environment where everyone felt comfortable and safe. It is important for everyone to be able to use the application and to be comfortable and able to focus on learning, so this should factor into the design of the application. Diverse testing participants will be essential to accomplishing an accessible product that aims to increase productivity for everyone. It is often difficult to predict how different people will react to products and therefore testing with diverse participants will benefit the product by gathering feedback from people with different backgrounds and experience with virtual reality, technology and education.

Due to the fact that 89.2% of nurses in the UK are female (Royal College of Nursing: Congress, 2019), developers of virtual reality nursing education platforms must take care to test and develop with their end users in mind. Focusing on providing a welcoming environment for everyone is essential, Gokalp (2019) identified that a tidy familiar environment made most people feel comfortable as shown in Figure 2. A good environment should be welcoming to everyone, in order to achieve this, good lighting is vital. Developers should ensure that the whole environment is well lit so that users can see everything in their surroundings. In the first environment of figure 2 the corners of the room are dark and difficult to see properly, this left users uncomfortable and this style of lighting should be avoided in educational apps where the users' comfort is a priority.

Figure 2. Bad Environment (left) compared to good (right). (Gokalp, 2019)

Implementation of Learning Techniques in Software

Coomans and Lacerda (2015) wrote about the many aspects of educational software that need to be considered during design. The article considers that the learning approach must be targeted to that of the people it is made for, i.e. when designing software for training nurses the software should use familiar learning approaches. However, past chapters identified that modern nursing education is not effective so new applications must improve upon the current methods. The article also evaluates creativity, exploration and self-analysis as effective teaching methods that improve understanding of the material. Implementing teaching methods such as experiential learning that require the user to use these skills in the software where appropriate should lead to more impactful learning.

Analyse Learning Methods

Experiential learning is a teaching method theorised by Kolb (1984), he posits that learning is the process whereby knowledge is created through the transformation of experience. For example, he argues that learning is most effective when experiences are being transformed into knowledge. When a learner has encountered an experience, they need to become directly involved with it in order to learn from it and make it an educational experience, engaging through either actions or reflection. Kreber (2001) suggests that field trips or trigger films can provide students with concrete experience upon which reflection learning can be based, however, the immersive qualities of virtual reality could make it an even stronger basis for experiential learning as it allows interaction between the user and the experience. For example, a virtual reality application can allow people to view medical conditions from different angles and get a convincing experience of how symptoms change over time, which should help them to recognise the symptoms in real patients when the time comes.

Gamification is defined as the use of elements usually associated with video games, e.g. point scoring or competition with others, to other areas of activity such as educational software. This can improve learning for students as shown in the study by Diaz-Ramires (2020), his educational game software led to a substantially greater passing rate compared to non-active players, and much higher engagement in problem-solving activities, surpassing even the level that the class required. This proves that gamification can be helpful, however as Berkling and Thomas (2013) found, gamification does not guarantee results. In their study they developed a game that was intended to be educational, they found that it was extremely inefficient at teaching and that students did not like having to work harder to get their information. This was a bad implementation of gamification, the idea should not be to create an educational game but to make the learning experience more rewarding using game ideology. Good implementations should focus on not creating unnecessary friction between the student and their learning. For example, including a points system can not only make the learning more fun but be useful for tracking progress and will not slow down the user's learning.

Positive reinforcement has been shown to be a strong motivator for people, it is used in lots of software to encourage users to continue certain behaviours. For example, video games use experience level systems to reward the user for their effort, which keeps them playing. Diedrich (2010) explains how this can be used to motivate students into continuing their work without them feeling tired or burnt-out. Applications can reward users with positive sounds and visual feedback when they get correct answers.

Currently, most education is teacher-led learning, however, the change to student-led learning in higher education can be more efficient as students can progress at their own pace (Turney, 2009). Simply put, student-led learning is where the student takes charge of the direction of their education, allowing them to learn in their own way. As Smith (2017) discusses this has proven benefits particularly in student satisfaction, with 90% of students preferring the student-led module to their other modules. Students also found it easier to focus on learning and it built self-awareness of their strengths and weaknesses. However, students often feel disassociated with learning due to challenges associated with overcoming specific obstacles. Group work could help overcome those obstacles.

Teamwork can be extremely beneficial to learning and student satisfaction, especially when learning in a new way, as shown by Heng-YuKu (2013). His research displayed that when making the transition to online learning, students benefited from working alongside others. As virtual reality will be a new medium to the majority of learners, utilising teamwork could be a good way to keep all of the learners interested, especially if there are fewer headsets than students. As Forrin and MacLeod (2017) concluded, using a combination of hearing, speaking and writing significantly boosts learning retention, therefore a good learning device would utilise all 3. Applying this to a virtual reality application could work as follows: a worksheet handed out to a group of students could describe the symptoms of a condition, which then should be described to a student using the headset, the students will then identify as a team the condition it matches and fill in the worksheet. The findings of Forrin and MacLeod (2017) are well documented and other pedagogical researchers have backed up the claims so using this knowledge in the design of educational software should be advantageous in increasing the productivity of learners and encouraging meaningful retention of information.

Successful education software will implement a range of pedagogical techniques to create an experience that maximises its benefits to learners. Techniques such as positive reinforcement can be simple to implement in certain apps for example quizzes but much harder in others, it is important to only implement techniques that do not obstruct the user's learning process. The use of teamwork as an educational tool could make an educational application more versatile, providing the option between group and solo work. It could also encourage learners to build soft skills such as communication, which is an essential skill in many fields including nursing.

Evaluation of Technology

Modern virtual reality technology comes in many form-factors including: mobile phone based, standalone and desktop based (Jung 2020). Each form factor has its own strengths and weaknesses as shown in figure 3, which should be taken into account when choosing the platform for your application.

Form Factor	Strengths	Weaknesses
Mobile phone based virtual reality e.g. Google cardboard	Low budget if users have phones already. (£5-£50)	Lowest hardware performance
	Portable	Limited to 3 degrees of freedom
	High potential audience	Low resolution
Standalone virtual reality e.g. Oculus Quest	Portable	Less powerful hardware than PC based virtual reality
	Straightforward setup and operation process.	Limited options
	Usually cheaper than PC based virtual reality (£300-£1000)	
Desktop PC based virtual reality e.g. Valve Index	Extremely high quality experiences possible	Expensive (£1000+)
		Tethered to a computer

Figure 3. A list of advantages and disadvantages to take into account when choosing a target platform for an educational application.

Developers must also choose a development platform to use. There are three main options to choose from: Unreal Engine, Unity and a custom solution. A custom solution requires a lot more time and skill as you are required to start from scratch and build custom build modules for every platform and cross-platform compatibility is much more difficult to achieve, however it does provide slightly more customization than the other two options, allowing for more potential in optimization and features given enough time.

Unity and Unreal Engine provide much quicker development allowing more time to be spent adding new features and optimising performance, it is very rarely advantageous to use a custom engine in modern real time 3D applications. Recent versions of Unity and Unreal Engine are very equally balanced, each having their own strengths and weaknesses, developers are likely to choose based upon whether they are better at C# (Unity's primary programming language) or C++ (Unreal Engine's primary programming language), however there are other factors to consider. For example, Unreal Engine is completely free for

commercial use, whereas Unity expects up to $1,800 per seat per year (Unity, 2021) in licence fees.

Conclusion

The results of this research describe methods in which to design educational software that focuses on creating a satisfying teaching experience for students, making learning simple by removing unnecessary obstacles and developing teaching methods that lead to better retention of information than traditional teaching methods such as books and lectures. The research particularly focused on opportunities for immersive virtual reality education, this growing medium has huge potential for simulator type experiences that can convey the sense of realistic conditions that are useful for students with no experience in the industry they desire to enter.

As discussed in figure 4, action feedback is vital in a virtual reality application, it shows the user when actions are completed and when implemented correctly provides positive reinforcement for the user's learning. In this way, developers can combine the findings of this paper such as the recommendation of implementing positive feedback and concentrating on good user experience design in order to create a product that is useful to the user.

Do's	Don'ts
Provide feedback for user actions such as button press sounds.	Create more work for learners in an attempt to make the experience more fun.
Design an environment in which all of the application's users feel comfortable.	Present information in unmanageable chunks.
Design your UI to be easy for users to navigate and understand.	Create controls that make the user onboarding experience unpleasant.
Choose the correct target hardware for your application.	

Figure 4 a description of required components in an educational virtual reality application.

Engaging the user in active learning is proven to boost learning progress and will create a memorable experience that will serve as good word of mouth for the educational software developed. If the software is designed to replace traditional methods of teaching, it is also important to allow users to quickly access information as they would be able to with a textbook. While this is often not a focus for immersive educational software, it can greatly benefit users to be able to search for information independently and learn at their own pace.

Analysis of the Requirements Specification

In order to determine the project requirements, we first had an interview with the potential client, in which he discussed his problem with the current methods of education in nursing and how in his professional opinion they are causing the high misdiagnosis rates discussed in the Terms of Reference.

Secondly, I developed a literature review, which allowed me to investigate teaching methods and educational apps. This helped me to develop requirements that the client approved of and that should improve the experience of learning in a virtual application by informing my design and development decisions. As appendix 2 shows the literature review had a huge impact on the requirements specification (i.e. requirements 2, 3 and 5) as many of the papers I referenced suggested that these were important to success when developing educational software.

I also developed a small prototype which allowed me to experiment with the idea and evaluate the potential of the concept. This prototype allowed me to gather further information from the client and directly led to requirement 6 being added to the specification (Appendix 2).

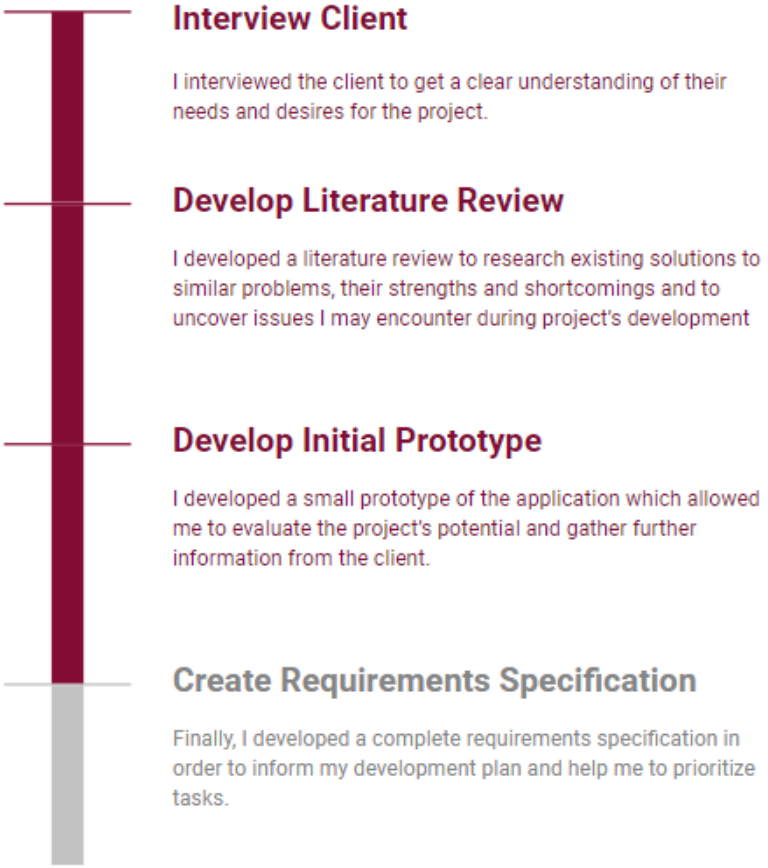


Figure 5. Process of the requirements synthesis.

I chose to develop in the Unity engine as I have lots of experience optimising programs in Unity, which will be important for a standalone virtual reality application. Also, the Unity engine is simple to use, has well-organized documentation and great third party virtual reality SDK support (Cosmina ISAR, 2016). I decided to avoid Unreal Engine due to the higher compilation times and worse performance of the editor. I considered a custom solution, developing my own engine, however, I decided that would take too long and have very few benefits.

I decided to use Scriptable Objects as opposed to a database to store condition information, this allowed the application to be used when not connected to the internet which I feel will greatly improve the user experience if used by touring educators as it will simplify the Quest setup process. This does create the limitation however that the application must be updated whenever new conditions are added or existing ones are updated, however this is a fairly simple process through the Oculus distribution portal. The use of Scriptable Objects allowed me to meet both requirement 1 and 7, which was a difficult design challenge.

Synthesis

Synthesis of Design

Before beginning development, I decided to create a class diagram to plan the design of the core systems of the application. This helped me to identify essential parts of the program and reduce coupling through creative design. For example, as Figure 6 shows I utilise Unity's Scriptable Object classes, they are designed to save data as an asset to use at run time and can be shared by multiple classes. They are used for storing condition information in my program as this makes them easy for anyone to edit. I also use the "Selected Disease" class to avoid using a singleton, it will hold the currently selected condition in the application. These scriptable objects decrease compile times and reduce dependencies in my program, which will be useful if an augmented reality or desktop app is requested by the client. Their use will also make it easier for me to add additional condition related features such as large projector screens with information based upon my research.

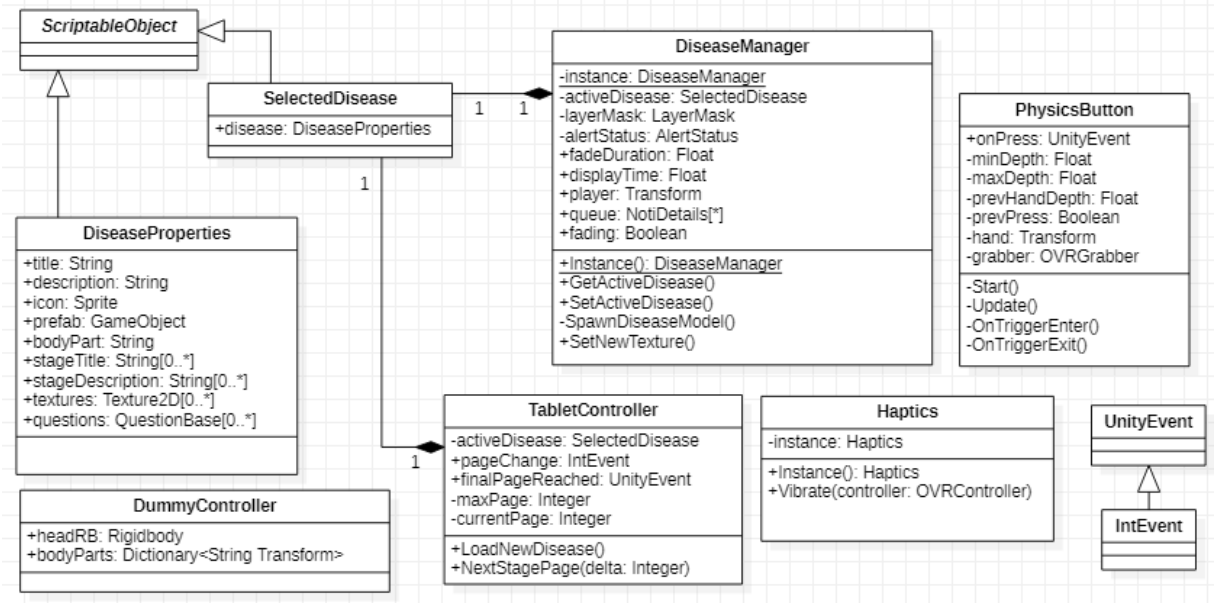


Figure 6. The first draft of a class diagram showing the design of the solution.

The systems in figure 6 I decided were the systems that I will develop first to ensure I have a working product before the project deadline. The "Disease Manager" class will control the active condition, manage the model and display the correct texture for each stage of the condition. The "Tablet Controller" will display the condition information and allow the users to change condition stages, which will be reflected by a change in the model's texture. Haptics will be a singleton that allows me to provide haptic feedback to either controller from any event. "Physics Button" is a simple push-to-activate button that can trigger any public script function.

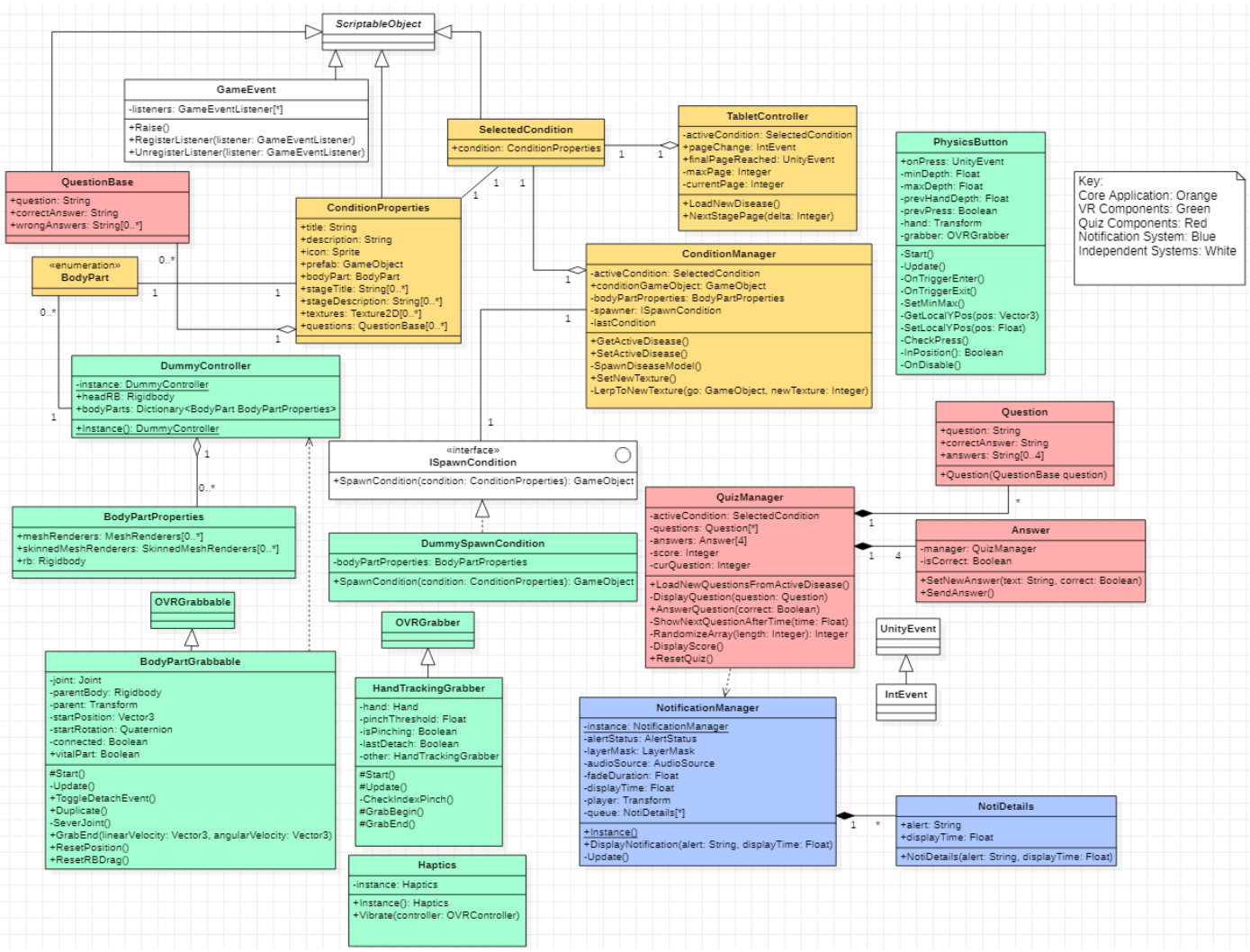


Figure 7. An updated draft of the class diagram to reflect Literature Review findings.

Upon completing my literature review I revisited my class diagram design to update it to reflect my findings. I decided to add a quiz to the application, which will allow users to test their knowledge. Whenever a user has finished reading about a condition, questions related to that condition will be added to the quiz, this provides feedback to the user that they are progressing when they finish their reading on a subject. The questions will be multiple choice and the quiz will be used to provide positive reinforcement to the user when correctly answering questions, via a green tick and positive sound cue. The points system will also add a game sensation that the user should be able to enjoy without it obstructing their learning, which my literature review identified as a priority when implementing such features.

I also changed the Condition Manager class, extracting the Spawn Condition implementation to an interface and removing the singleton in order to make the class reusable. Now the class should be used for any instance of a condition that is spawned. This can work if multiple different conditions are spawned or multiple copies of the condition have to be controlled. It can progress them through stages together or separately as that is controlled by the

TabletController. This clever design will benefit me when developing for multiple platforms, further developing this platform or if other developers are added to the project as it lowers coupling and increases cohesion.

The “Hand Tracking Grabber” class (green in Figure 7) overrides the base OVRGrabber class from the Oculus Integration allowing me to develop gesture recognition for grabbing, while also allowing it to share the grabbable object functionality of the Oculus SDK’s controller based grabbing class. This should remove the need for controllers making the application simpler to use as my literature review suggested.

I’ve also added a flexible notification system that will be used to alert the user of changes, for example when new questions are added to the quiz area. This is completely independent of any other system in the solution so can be reused with any alerts that may be required in future additions to the project.

Synthesis of Implementation

Initial Development of the Core Application

To begin my implementation I created a Unity project and imported the Oculus SDK, this allowed me to set up the project to build for the Oculus Quest. I utilised the OVRPlayerController prefab, configuring it with my own settings to allow the user to walk around the scene. I also created a basic hospital room to be the user’s environment for early testing.

The condition properties class is built upon the scriptable object class, which will allow me to create an object to store the details of any condition I wish to represent before runtime. This class was designed to contain all of the condition information with no functionality so that it can be used by any part of the application that needs condition information. I also chose to store the currently selected condition as a reference in a separate scriptable object class as this allowed the condition to be set by one function but accessed by as many classes as needed.

As a result of the initial discussions with the client keeping code reusable has been a priority throughout this project. There is a lot of potential to grow the relationship with the client and develop other apps around the same idea such as an augmented reality version of the same app. The development of such an app can be greatly simplified by separating classes correctly, reducing dependencies and keeping low coupling and high cohesion as a priority during development. As shown in appendix 3, I use several methods to reduce class dependencies, for example, instead of implementing the spawn method in the condition manager class, I extracted it to a separate class and called it via an interface. This separation will allow the condition manager class to be reused, in for example an augmented reality app,

to control the condition model but the scene will not require a “dummy” to spawn on as the virtual reality application does.

The tablet controller class was developed to display the condition information, breaking it up into stages in order to reduce the amount of information that the learner must take in at one time. This was encouraged by the research from my literature review which suggested that an advantage of software over conventional book-based learning was that the user can be presented with smaller chunks of information making it easier to digest. The stages can be progressed using buttons, which provide feedback to the user and keep them engaged by requiring interactions to progress.

The condition manager creates and controls the 3D model which displays the symptoms of the currently selected condition. This model is the main feature of the application which was pitched to us by the client, he suggested that learners seeing realistic models of conditions would provide a better learning experience and help them retain detailed knowledge of the symptoms reducing misdiagnosis in the long run. This theory was backed up in my literature review which showed that a memorable experience provides a strong basis for learning. The condition manager class also controls the texture on the model, subscribing to the tablet controller’s page change event allows it to update the texture to reflect the tablet’s current stage.

An interesting and useful skill I developed while creating this application was writing and understanding shader code, which is a particularly difficult part of Unity development but can be very useful. In this project I used it to develop a material capable of blending smoothly between 2 textures. This made the application feel much more polished and professional as the transition between condition stages is no longer very abrupt, instead showing that there is a progression over time. The full shader file is available in appendix 5 titled “Blend 2 Textures Shader”, but a simple overview is available in figure 8, the shader interpolates between the colour values of both textures for each pixel based upon the lerp value. It also takes into account lighting, and shadows.

```
void surf (Input IN, inout SurfaceOutputStandard o)
{
    fixed4 c = lerp(
                        tex2D(_MainTex, IN.uv_MainTex),
                        tex2D(_SecondaryTex,  IN.uv_SecondaryTex),
                        _LerpValue
                    );
      o.Albedo = c.rgb;
      o.Metallic = _Metallic;
      o.Smoothness = _Glossiness;
      o.Alpha = c.a;
}
```

Figure 8. The Surface function of the Blend 2 Textures Shader from appendix 5

The body part grabbable class was developed to allow the user to grab and move body parts. This class inherited from the base OVRGrabbable class in the Oculus SDK and added functionality to allow the parts to attach and detach from the skeleton, so that they can be viewed from any angle and to duplicate parts to allow different conditions and stages to be compared.

Reactions to the Literature Review & Client Feedback

The quiz section was added as a result of feedback from my boss and the suggestion in my literature review that gamification can significantly benefit the user's learning experience. I wanted to add a way to test the user's ability that doesn't obstruct their learning experience. The quiz uses scoring to reward users and show progression as they learn and aim to improve their score, this is a popular gamification technique that makes learners want to learn and feel rewarded for their efforts.

Figure 9. Screenshots showing the change in environment.

My literature review revealed the importance of the environment in virtual reality, which is why I decided to review the environment of the application, which I identified was currently boring and cramped, which might make users feel uncomfortable and uninspired. This led to the

application getting a much needed environment change as shown in Figure 9. The new environment is much better, when developing it I focused on creating warm lighting and creating an open space that makes users feel comfortable and allows them to focus on learning.

The notification manager class is used to display notifications to the user. When a notification is sent a sound is triggered to alert the user and then when they are looking at the tablet the notification message will fade in for a specified amount of time.

Upon receiving feedback from the potential client and my boss I decided to begin implementing hand tracking. I left this till late in the project as I identified it as of low importance, however when testing the client and I found that the biggest problem with the application in its current state was that onboarding new users was difficult. Describing button bindings and controls when the user is new to virtual reality is complicated so removing buttons and moving towards a touch interface should improve the user experience. I created the hand tracking grabber class, which inherits from the OVRGrabber class and is designed to trigger it's grabbing functionality without requiring button presses, instead it checks for a pinch between the index finger and thumb. I chose this method of interaction as it is ideal for accurate tracking (Wilson, 2006) and easy for users to understand.

Synthesis of Testing

Approach to Testing

When deciding upon my testing strategies for the project I decided I wanted to prioritise interviewing the tester as it would allow me to get as much explanation as possible for their feedback. The testing participant was not a part of the project but a PHD candidate at Northumbria University, a physiotherapist and a researcher focusing on digital health technologies and wearable computing. His feedback is particularly valuable as he is a medical expert and is well informed in regards to using technology as a medical tool.

Prior to the interview I circulated a video of me using the application and demonstrating the system's features to the test participant, allowing them to review the use of the application before the meeting. Then during the meeting, we re-watched the video while I pointed out features and I gave the participant the opportunity to ask questions and scrub through the video to aid in their feedback. I also asked them a series of prepared questions upon topics which I wanted to receive feedback such as the application environment and usability of the experience.

In order to get the most value from this opportunity I scheduled a second interview with the testing participant, allowing me to react to their feedback, making many of the suggested improvements and verify that my responses improved the application in their professional opinion.

Limitations of Testing

Coronavirus was hugely limiting in my testing, preventing me from allowing users to try the application due to safety concerns and legal restrictions. Instead I recorded a video of myself using the application and showed it to the tester, providing commentary over my actions. I was very skeptical of the effectiveness of this testing solution, however it did provide me with some very good actionable feedback.

The main issue with this form of testing is that many of the important points of testing such as the user onboarding experience are difficult to judge without being able to experience it for yourself. The only remedy to this is being extremely critical in my own evaluation of the solution, which I have been throughout the development of this project, and when asked my tester couldn't identify any issues.

Testing Response

The tester did, however, suggest that I remove the need to detach body parts from the skeleton, instead instantiating two models; one attached and one separated from the skeleton. The reason for this was that some people might find the action uncomfortable, I found this an interesting point of view and saw no reason not to implement this suggestion into the application. This change was impressively easy to make due to the use of an interface to spawn a condition for the DiseaseManager Class, requiring just 1 new class implementing the SpawnCondition interface as shown in figure 10 and 11 and Appendix 6 (the Rotator Spawn Condition class).

```
GameObject SpawnCondition(conditionProperties) {
    var model;
    if(conditionProperties != null){
    model = Spawn(conditionProperties.prefab, transform)
    model.localPosition = Vector3.zero
    model.layer = grabbableLayer

    model.AddComponent(Rigidbody)
    model.rigidbody.kinematic = false

    model.AddComponent(PartGrabbable)
    }
    return model;
}
```

Figure 10. Psuedocode that describes the class required to control the instantiation of a new condition.

Figure 11. Screenshots showing the 2 separate models and how they are spawned, both can be grabbed and reflect the changes of different stages of the actively selected condition.

Also, the testing participant suggested that I modify the tablet that displays condition information, replacing it with a binder similar to that which you would find at the end of a patient's bed in a hospital ward.

"To make the application a better simulation of a ward environment it could be helpful to display condition information in a folder, allowing learners to get used to the format they will be expected to use in a ward environment"
- Anonymous testing participant response.

Explaining this suggestion they said that it would prepare learners for the actual experience of being on a hospital ward and that this could make the application even more useful as a tool to allow learners to experience the job they are studying to achieve in a risk-free environment. While this is not the main purpose of the application, it could make it useful as a University open day tool to attract learners to medical studies courses. Also, it drastically improves the appearance of the application as the tablet is one of the first things you see in the application and the most time-consuming part, yet it was not very visually appealing at the point of testing. This was definitely a good improvement as the appearance of the application is especially important when pitching the product to clients. The improvements also allowed me to improve the formatting of the information presented and include images in the stage descriptions as shown in Figure 12.

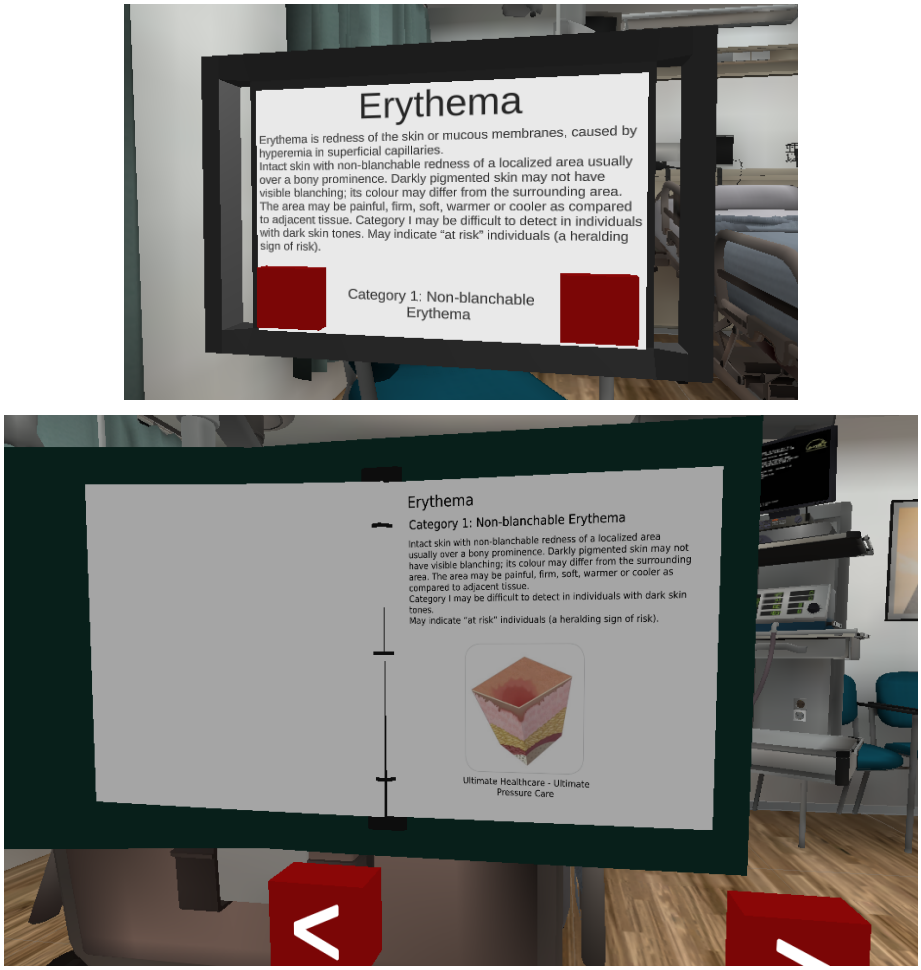


Figure 12. Screenshots showing the change in the way information is delivered.

After carrying out testing I decided that the application should support both hand tracking and controllers, however, that hand tracking was suitable for most use cases. Hand tracking is accurate and the easiest method of input for the vast majority of circumstances, however, it performed poorly if the room was poorly lit and for differently-abled users controllers can afford them greater accuracy and more controls for abilities such as height control. Hand tracking also has poor support for people who are missing digits (i.e. fingers). While this will affect a very small minority of users I feel that it is important not to exclude any group of learners from access to educational material. When asked, my tester believed hand tracking to be the most intuitive and that haptic feedback was not too important due to the sound and visual feedback also provided in the application.

The tester also suggested that I implement a human patient that shows the conditions for users to diagnose, for example, a man sitting in the bed, that shows the heel of his foot. While I think this is a good idea, it would be expensive to implement as I would have to purchase a customizable animated person model and it would take a lot of time to implement, also requiring more textures to be designed for each condition. For this reason I decided not to implement it in the solution yet, it is however a highly desirable option for if the project gets

funded as future company work. This would provide the application a better method of quizzing than I currently have implemented and might provide a more accurate understanding of what the user has learned.

After evaluating the changes suggested by my tester and implementing the ones that I deemed would be beneficial to the application's experience I held a second meeting with him to review the changes I made. Overall the tester's response was extremely positive, expressing that the implementation of the binder was a significant improvement in their eyes, being much clearer and making the information easier to process.

The tester also suggested that I move the new condition spawning system from the table to the hospital bed, stating that they think this is where the learner would expect to find it. While I understand their reasoning I disagree with the idea as I think it will make the smaller body parts such as the foot hard to find for first time users due to the higher concentration of objects in that area. Also, that area of the environment has several objects that the users will have to navigate around, this could become an issue for users not familiar with similar systems as the simulated movement is often new to them.

Employer and CNTW Feedback

My employer and a nursing specialist educator employed at Cumbria, Northumberland, Tyne and Wear (CNTW) NHS Trust and the original pitcher of this application, had the opportunity to actually use the application. They have responded extremely positively to the application's development. The CNTW nurse specifically pointed out that the smooth transition between condition stages was impressive and particularly points to the detail visible in the solution when presenting the condition. This is a key feature of the application and will be a primary point of focus when adding new conditions. My employer intends for me to resume development of the application as I continue my full time employment after completing university, and we are now considering pitching it as a product of Boxmodel to be licenced to educational institutions such as Health Education England, NHS. My anonymous tester feedback also suggested this application could now be used for on the job placement training after an Oxford study has now approved the use of virtual reality replacing clinical placement hours (Pottle, 2021).

I received stakeholder feedback from my employer based upon my development of the project (Appendix 4). They particularly appreciate my focus on architecture design as it is an important skill to have when developing large projects.

"develop-'ed' *a solution capable of rapid adaptation to client requirements"*
Quote from the Stakeholder feedback.

As the quote describes, the flexible architecture of the project has allowed me to build a generalised solution that can quickly change to meet the requirements of a client, while allowing code reuse and minimal extra effort.

Evaluation

Evaluation of Product

Summary of the Products Strengths and Weaknesses

The final product of this assignment is an educational application that successfully employs the learning techniques uncovered in my literature review. I believe that, as my testing suggested, the application has the potential to be very useful for medical education providing an engaging experience that boosts detail recollection and is more enjoyable for learners. In this chapter I will review its effectiveness and if I successfully met the aim of developing an educational tool, that helps to reduce cases of misdiagnosis in the NHS.

A major weakness in my project was that it lacked a definitive direction, as a company we have been limited in our virtual reality outreach to potential clients by coronavirus compared to previous years. This has led to us not being able to pitch the product as effectively to clients and get it in front of as many eyes as possible, because of this I refused to limit the project scope to a single cohesive product, instead I developed a framework which can easily be adapted into a successful application. This has the advantage that the codebase is very generalised and could be used to develop many different applications in the future, however it has led to a product that would not currently be useful in the hands of a client. While this was an intentional design decision taken to improve the usefulness of the project to my employer, the application could appear stronger in pitches if it were more fleshed out with more conditions.

A minor drawback of my project is the limited use due to only a single condition being present in the application. The reason I did not add more conditions was that my employer has to pay for new textures to represent each condition and I do not believe that is cost-effective prior to locking down a client for the project. However, the project has been built to allow new conditions to be added in just a few minutes, meaning that if the project is funded and we get the textures drawn we can quickly implement new conditions and more time can be spent developing new features.

When we are able to secure a client for the project we will be able to target more key features, for example, if the client decides the application will be most useful to them in a classroom environment, where students can use it repeatedly I would strongly recommend adding an achievements system to reward the user for their progress. My research suggested that gamification can be very helpful when encouraging user learning and I think achievements would be a fun way to encourage interactivity. The achievements could earn learners points and badges, which would create friendly competition between students. However, an achievements system would be pointless if the application is only to be used by students when a touring educator visits for a single session per year, for example, so building it ahead of securing a client would be a waste of resources.

What I believe my application does exceptionally well is provide a strong framework for an educational application that we can sell to a client, my employer fully agrees with this assessment and we are currently exploring options for how best to sell the application. The framework I've provided showcases the possibilities of a virtual reality education application and will impress potential clients more than any simple PDF overview could as discussed in my stakeholder feedback (Appendix 4).

Product Requirements

My application comfortably meets all of the requirements laid out in appendix 2, however, I believe there might be room for improvement in requirement 3. While the appearance of the condition is very impressive, for some conditions including erythema the sense of smell can be used to help diagnose it. Since smell cannot be conveyed in virtual reality (without the need for additional hardware), this is currently ignored in the application. However, I would like to consider adding a simple graphic to convey that the sense of smell can be used to assist diagnosis, but should be designed in a way that doesn't remove the sense of realism from the application.

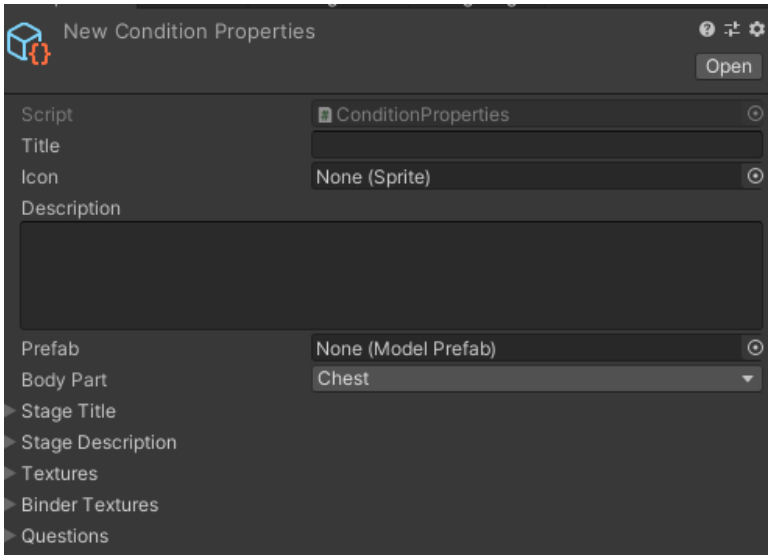


Figure 13. Adding a new condition form

The application meets requirement 1, allowing easy addition of new conditions but just submitting the information as seen in figure 13. This allows new conditions to be added easily and prevents the application from requiring an internet connection to be used, which may be important to a client.

I also removed any need to learn button bindings for this project by implementing hand tracking. This allows it to meet requirement 2 and 6 as users can easily navigate the user interface without virtual reality experience as all of the buttons are visible and intuitive.

Requirement 8 required me to take extra steps such as cutting high polygon models from the scene such as the window blinds. I identified these issues using Unity's overdraw feature

which identifies the highest processing cost parts of the scene. I also baked the lighting for static meshes, which reduces draw calls and improves performance drastically.

Reviewing my aims and objectives for the project, I think that I was unable to prove that my application will effectively reduce cases of misdiagnosis. However, I think that given enough time my research for this project shows that it has the potential to vastly improve medical education if given the opportunity as the objective I to investigate teaching methods was successful and my literature review shows that the methods implemented in the application have pedagogical backing.

Future Work

In order to boost the usefulness of the quizzing component to the application, my employer and myself are evaluating implementing an artificial intelligence component to allow the application to detect weak points in the user's knowledge and assist the user to improve their learning technique within the application. This machine learning system will run in the cloud as to not affect the performance of the application on the low powered Oculus Quest headsets, so will require an internet connection and data to be passed to a Microsoft Azure backend.

The Azure cloud system will review the steps you take in the application and data such as how long you spend reading about certain conditions. Then it will compare this information to the answers you provide in the application's quiz, by inputting this data into a machine learning algorithm, we hope to be able to improve the student's learning by predicting their weak spots as discussed by Hussain (2019). They recommend that using either an Artificial Neural Network or a Support Vector Machine an application can measure the difficulty each student experiences during each session. Using this information the system could improve the next session for each individual student by tailoring it to their difficulties and feedback to tutors if lots of students found a session unusually difficult. The use of quizzing in the application will help to train the algorithm as we will have solid score based data upon which to compare and value each session.

Evaluation of the Project Process

Evaluation of the Initial Approach

This chapter reviews the process I used to develop the project, evaluating the decisions I made and exploring changes I would make in future projects to improve the development cycle or the resultant product. I will also compare the process to the aims and objectives I set at the start of the project related to the development process, specifically developing an informed requirements specification, learning better software design for users and architecture and evaluating the testing process I used.

I developed this solution in the Unity game engine, Unity is named for its proficiency in developing a single solution for many platforms including mobile, desktop, console and it is one of the most popular virtual reality development solutions. While Unity is a great solution for developing virtual reality software, the primary reason I chose it for this application is its potential to support future development for example, if the client wants an augmented reality version of the application I could share the same codebase for both platforms of the application.

The only viable alternative to Unity for a project of this size is the Unreal engine, the advantage to using this engine is that it is entirely free for commercial use, unlike Unity which requires a subscription. However, I decided against using it as it requires C++ programming which I do not know, whereas I have a lot of experience with Unity and the C# programming language. This project is already quite large for a single person to develop, which is why I decided that trying to learn C++ and the Unreal engine alongside this would be too much work and in hindsight, I think that I made the correct decision.

The hardware I decided to target was the Oculus Quest, the key influence in this decision was that after my first experience with standalone virtual reality I knew it was the future. Since the project started the Oculus Quest 2 launched to an outstanding reception, managing to outsell tethered headsets like the Oculus Rift that have been available for five years in just 5 months (TechRadar, 2021). I think that this shows that stand-alone virtual reality has the biggest potential for mass adoption and should be the standard choice for commercial application development as it removes barriers such as expensive desktop computers and setup time. The application I have developed fully supports both the Oculus Quest 1 and 2, and because of my decision to develop in the Unity engine will be able to be updated to support any future hardware releases making it more valuable in the long term to the client.

Evaluation of Development Process

My development process inside of Unity was quite efficient, I managed to quickly learn the Oculus integration library due to my experience developing inside of Unity on my placement year. Also, my UML diagrams helped me significantly as I was able to consider and reduce dependencies ahead of developing systems. This helped me to meet requirement 10, developing a system with low coupling and high cohesion.

The biggest issue I had during the development process was debugging using the Oculus Quest headset, while Unity has excellent debugging tools I had several issues that only occurred when wearing the headset, this was fixed when Oculus released “Oculus Link” allowing me to use the headset in editor. However, in future projects I would recommend spending some time developing editor tools that allow for faster and more seamless testing in play mode without virtual reality, it would not have required much effort due to the system design and would have saved lots of times debugging issues as errors are easier to track in the Unity editor than via the Android Debug Bridge.

Evaluation of Report Process

Overall, I think that the project development process was very strong, however I definitely struggled to produce an effective literature review. Thanks to the help of my supervisor I did eventually complete a literature review that benefited the development of the project but it could have progressed much quicker if I had more experience writing similar reports or had spent time researching what is expected to belong in the document. The literature review completed, successfully met the standards of my project aim, identifying good teaching methods that I could implement in the application.

The literature review also allowed me to make informed decisions based upon evidence to improve the application, which I believe made a significantly positive impact on the finished product. For example, the environment design improvements I made as a result of the research I documented in the literature review significantly improved the experience of the application in my own opinion, and my testing participant commented that the environment was “welcoming and convincing of a real ward experience”.

I also feel that I succeeded in my aim to learn better design of software, due to the research completed for my literature review and during development. I was able to compare design guidelines from different companies such as Google and Apple and make an informed decision as to how I should implement them in my application. The application I have developed is user friendly following many of the virtual reality industries best practices such as accessible controls, varied input methods and artificial locomotion control. I also produced report chapters explaining my software architecture decisions, which have led to a system capable of rapid iteration and easy onboarding of new developers.

Another weakness in my project process was the testing, which was severely limited by coronavirus restrictions. I did not plan for the possibility that we would enter another national lockdown as the government at the time insisted it would not be necessary. Due to this my testing suffered, being unable to allow the participant to use the application as my project aims required.

Although I was unable to fully complete my aim, I instead demonstrated the use cases of the application through a video of the application in use to a testing participant. This allowed me to gain the perspective of a qualified medical professional to inform my development. While I did gain valuable insight from this testing, I will be pursuing further testing of the application when restrictions are lifted to gain a better understanding of its potential users and their needs.

The risk assessment table I completed in my Terms of Reference was incomplete and not fit for purpose. Upon revisiting it I should have added the extra situations described in figure 14 to ensure that I was prepared for more situations.

Description	Severity	Likelihood	Action
Coronavirus could prevent me from attending University and testing the final product.	Low	High	All work stored in the cloud so that it can be accessed from home. All required hardware is my own so access to the University is not required. Testing should follow rigorous health and safety protocols, including mask requirements, hand sanitiser and headset sanitization. If unable to be completed in person, testing will have to be compromised. I will critically self assess the solution and record video footage of myself using the application for a tester to review.
Hardware breaking, losing access to testing and possibly the project.	High	Low	I have access to multiple VR headsets and computers. Backups of the project will be stored on Github and the report on Google Drive.
Inaccurate information presented in the system could cause harm.	High	Low	All information will be provided by NHS professionals and reviewed before release to a client.
Unable to source clients prior to completion of the project.	Low	High	If a client is not found before the project finishes, my employer (Boxmodel) will be treated as the client as the project is being built for them anyway.

Figure 14. Risks that should have been included in the risk assessment table

Conclusions and Recommendations

Upon finishing this module I will continue development of this project alongside other projects while we at Boxmodel continue our search for clients, several are in discussion with my employer to purchase or fund its continued development. Securing a client will allow me to improve the requirements specification for the project and add features such as the achievements system discussed earlier as the use case can be narrowed down to specific situations. The clever system design allows for new components to be added to the application with ease, as shown when I responded to testing feedback.

To improve the usefulness of the application I also intend to develop a mobile augmented reality component, which will allow for greater accessibility to the application as people will be able to download it on any modern smartphone, removing the need for headset distribution. As displayed in appendix 3 virtual reality components are only a small section of the application, the rest can be reused in the augmented reality app, as this was an anticipated proposition from the start. This should simplify development as the main features are already complete, just the augmented reality interaction systems must be developed.

While I met every one of my requirements in the specification, there are still steps required before this application should be delivered to a client. For example, I would like to improve the responsiveness of the buttons. As new standards for virtual reality applications are set, many new applications are opting to show interaction is possible in creative ways. As shown in figure 15, buttons can provide better accessibility and a more polished or premium experience than I currently offer in the application by being more responsive to the user's actions.

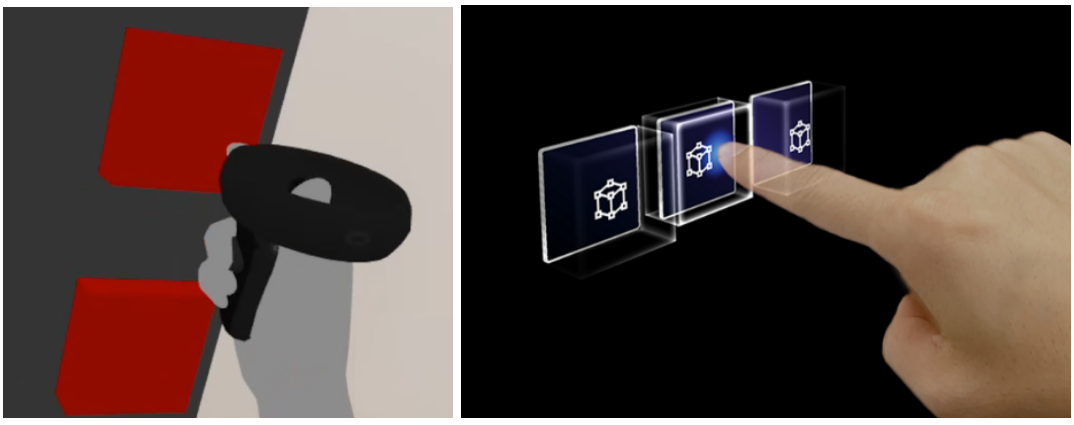


Figure 15. My application buttons compared to the Mixed Reality Toolkit, which uses glows and holograms to show to a user that a button is interactable. (Park, 2019)

I would recommend to anyone developing a virtual reality application, that they particularly focus on creating polished interactions. Small effects in virtual reality such as the texture blending shader I discussed earlier make a lasting impact on users, generally creating a better user experience. Polish can also lead to more intuitive design, as with the buttons in

figure 15, effects should be used to lead the user towards interactions you want to encourage.

My final aim for the project was to complete a critical evaluation of the product and its development process. In order to do this the previous two chapters of my report assessed the quality of the solution I developed and the process by which I developed it. I used testing feedback to value the effectiveness and potential of the application and discussed the issues and challenges overcome during its production.

Another consideration that I would like to test going forward is that I do not believe that virtual reality is a hugely important part of the project despite what my literature review suggested. My tester feedback suggested that an augmented reality version of the application could also be very helpful and I think that the key to the project is actually the simulation and interactivity it can provide. For this reason I would like to test the application working on other platforms, the architecture was designed to support it easily so with a week of free time I could develop it quite easily.

If I was to provide advice to anybody developing educational virtual reality software, I would encourage thorough planning when adding features as this will benefit you whenever features are updated, improved or added. Separating systems into individual parts, while taking longer to initially implement, will benefit you over long project durations, not only saving time but also allowing bigger and better systems to be built. Also, taking the time to document and review research can be extremely helpful to developing an application that provides users with the best experience possible and completes its purpose.

Bibliography

Introduction

Oculus (2020)
How Hilton Uses Oculus for Learning & Development | Oculus for Business
Available at:
https://business.oculus.com/case-studies/hilton/
(Accessed: 04/05/21)

Product Overview

NHS Resolution (2018)
NHS Resolution Freedom of Information F/3216
Available at:
https://resolution.nhs.uk/wp-content/uploads/2018/09/FOI_3216_Misdiagnosis.pdf
(Accessed: 14/02/2021)

Literature Review

Huang, C. L., Luo, Y. F., Yang, S. C., Lu, C. M., & Chen, A.-S. (2019)
Influence of Students' Learning Style, Sense of Presence, and Cognitive Load on Learning Outcomes in an Immersive Virtual Reality Learning Environment
Available at:
https://journals.sagepub.com/doi/abs/10.1177/0735633119867422
(Accessed: 03/11/2020).

Jeffrey Michael Franc-Law, Pier Luigi Ingrassia, Luca Ragazzoni and Francesco Della Corte (2015)
The effectiveness of training with an emergency department simulator on medical student performance in a simulated disaster.
Available at:
https://www.cambridge.org/core/journals/canadian-journal-of-emergency-medicine/article/effectiveness-of-training-with-an-emergency-department-simulator-on-medical-student-performance-in-a-simulated-disaster/D1156F8D878962C50A346B2059CF20EC
(Accessed: 04/11/2020)

Arjuna I. Ranasinghe, Diane Leisher (2009)

The Benefit of Integrating Technology into the Classroom
Available at:
http://www.m-hikari.com/imf-password2009/37-40-2009/ranasingheIMF37-40-2009.pdf
(Accessed: 16/12/2020)

Peter Bradford, Margaret Porciello, Nancy Balkon, Debra Backus (2007)
The Blackboard Learning System: The Be All and End All in Educational Instruction?
Available at:
https://journals.sagepub.com/doi/abs/10.2190/X137-X73L-5261-5656
(Accessed: 18/12/2020)

Roumen Vesselinov, John Grego (2012)
Duolingo Effectiveness Study
Available at:
https://static.duolingo.com/s3/DuolingoReport_Final.pdf
(Accessed: 18/12/2020)

Jacek Uziak, M. Tunde Oladiran, Edmund Lorencowicz, Kurt Becker (2018)
Students' and Instructor's Perspective on the use of Blackboard Platform for Delivering an Engineering Course
Available at:
https://digitalcommons.usu.edu/ete_facpub/237/
(Accessed: 15/12/2020)

N.W. Coppola, R. Myre (2002)
Corporate software training: is Web-based training as effective as instructor-led training?
Available at:
https://ieeexplore.ieee.org/abstract/document/1029957
(Accessed: 04/11/2020)

Mustafa Hussein, Carl Natterdal (2015)
The Benefits of Virtual Reality in Education: A Comparison Study
Available at:
https://gupea.ub.gu.se/bitstream/2077/39977/1/gupea_2077_39977_1.pdf
(Accessed: 21/12/2020)

Park, E. L., & Choi, B. K. (2014)
Transformation of classroom spaces: traditional versus active learning classroom in colleges.
Available at:
https://link.springer.com/article/10.1007/s10734-014-9742-0
(Accessed: 21/12/2020)

Don Allison, Larry Franklin Hodges (2000)
Virtual reality for education?
Available at:
https://dl.acm.org/doi/pdf/10.1145/502390.502420

(Accessed: 04/11/2020)

Liam Murray, Ann Barnes (1998)
Beyond the “wow” factor—Evaluating multimedia language learning software from a pedagogical viewpoint
Available at:
https://www.sciencedirect.com/science/article/abs/pii/S0346251X98000086
(Accessed: 06/11/2020)

Ayfer Gokalp, Tara Franz (2019)
Women in VR User Research, Oculus Connect Talk
Available at:
https://www.youtube.com/watch?v=Oo5cC5S3EGA
(Accessed: 09/11/2020)

Kolb, D.A. (1984)
David A. Kolb
Experiential Learning: Experience As The Source Of Learning And Development
Publisher: Prentice-Hall, ISBN: 0132952610
Available at:
https://www.researchgate.net/publication/235701029_Experiential_Learning_Experience_As_The_Source_Of_Learning_And_Development
(Accessed: 21/01/2021

Carolin Kreber (2001)
Learning Experientially through Case Studies? A Conceptual Analysis. Teaching in Higher Education.
Available at:
https://srhe.tandfonline.com/doi/abs/10.1080/13562510120045203#.YAqrf-j7SHs
(Accessed: 21/01/2021)

Jose E. Naranjo, Diego G. Sanchez, Angel Robalino-Lopez, Paola Robalino-Lopez, Andrea Alarcon-Ortiz and Marcelo V. Garcia (2020)
A Scoping Review on Virtual Reality-Based Industrial Training
Available at:
https://www.mdpi.com/2076-3417/10/22/8224/htm
(Accessed: 10/11/2020)

Oculus For Business (2020)
How Hilton Uses Oculus For Learning & Development
https://business.oculus.com/case-studies/hilton/
(Accessed: 17/01/2021)

Yong Min Kim, Ilsun Rhiu & Myung Hwan Yun (2019)
A Systematic Review of a Virtual Reality System from the Perspective of User Experience
Available at:
https://doi.org/10.1080/10447318.2019.1699746
(Accessed: 10/12/2020)

Kauhanen, O., Väätäjä, H., Turunen, M., Keskinen, T., Sirkkunen, E., Uskali, T., Karhu, J.

(2017)
Assisting immersive virtual reality development with user experience design approach.
Available at:
https://doi.org/10.1145/3131085.3131126
(Accessed: 10/12/2020)

Stefania Serafin, Giovanni Serafin (2004)
Sound Design To Enhance Presence in Photorealistic Virtual Reality
Available at:
https://smartech.gatech.edu/bitstream/handle/1853/50913/SerafinSerafin2004.pdf
(Accessed: 10/12/2020)

Jessie Y. C Chen (2019)
Virtual, Augmented and Mixed Reality. Multimodal Interaction Cham, Springer International Publishing
Available at:
https://link.springer.com/chapter/10.1007/978-3-030-21607-8_3
(Accessed: 14/12/2020)

Narges Ashtari, Andrea Bunt, Joanna Lynn McGrenere, Michael Nebeling, Parmit K. Chilana (2020)
Creating Augmented and Virtual Reality Applications: Current Practices, Challenges, and Opportunities
Available at:
https://dl.acm.org/doi/pdf/10.1145/3313831.3376722
(Accessed: 14/12/2020)

Merijn Kersten (2018)
Designing user interaction in VR
Available at:
https://medium.com/@demerijnkersten/designing-user-interaction-in-vr-eb091bc0bbc1
(Accessed: 14/12/2020)

Bojan Kraut, Jelena Jeknić (2015)
Improving education experience with augmented reality (AR)
Available at:
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7160372
(Accessed: 15/11/2020)

Royal College of Nursing: Congress (2019)
Debate: Male nurses
Available at:
https://www.rcn.org.uk/congress/what-happened-at-congress-2019/Male-nurses#:~:text=There%20are%20currently%20over%20690%2C000.jobs%20in%20the%20United%20Kingdom.
(Accessed: 14/01/2021)

Stéphanie Coomans, Gilberto Santos Lacerda (2015)

PETESE, a Pedagogical Ergonomic Tool for Educational Software Evaluation
Available at:
https://www.sciencedirect.com/science/article/pii/S2351978915008963
(Accessed: 15/11/2020

Jennifer L. Diedrich (2010)
Motivating Students Using Positive Reinforcement
Available at:
https://digitalcommons.brockport.edu/cgi/viewcontent.cgi?referer=https://scholar.google.co.uk/&httpsredir=1&article=1008&context=ehd_theses
(Accessed: 16/11/2020)

Turney, C. S. M., Robinson, D., Lee, M., & Soutar, A. (2009)
Using technology to direct learning in higher education. Active Learning in Higher Education.
Available at:
https://www.tandfonline.com/doi/abs/10.1080/03098265.2013.811638
(Accessed: 22/01/2021)

Jenny Díaz-Ramírez (2020)
Gamification in engineering education – An empirical assessment on learning and game performance
Available at:
https://www.sciencedirect.com/science/article/pii/S2405844020318156
(Accessed: 16/11/2020)

Kay Berkling, Christoph Thomas (2013)
Gamification of a Software Engineering course and a detailed analysis of the factors that lead to its failure
Available at:
https://ieeexplore.ieee.org/abstract/document/6644642
(Accessed: 16/11/2020)

Carolyn Smith (2017)
The flipped classroom: Benefits of student-led learning
Available at:
https://journals.lww.com/nursing/Citation/2017/04000/The_flipped_classroom__Benefits_of_student_led.7.aspx
(Accessed: 18/11/2020)

Heng-YuKu, Hung Wei Tseng, Chatchada Akarasriworn (2013)
Collaboration factors, teamwork satisfaction, and student attitudes toward online collaborative learning
Available at:
https://www.sciencedirect.com/science/article/pii/S0747563212003597?casa_token=rASgn1LLAiAAAAAA:qv6-mDDjOFzT6gTYDx76dC1ZVKS9G1hbs8gUQIn7z8ea2XkPNSEJNMRsMQEzy7jHCUpYwfHaew

(Accessed: 01/12/2020)

Noah D Forrin, Colin M MacLeod (2017)
This time it's personal: the memory benefit of hearing oneself
Available at:
https://pubmed.ncbi.nlm.nih.gov/28969489/
(Accessed: 01/12/2020)

Timothy Jung, M. Claudia tom Dieck, Philipp A. Rauschnabel (2020)
A Classification of Virtual Reality Technology: Suitability of Different VR Devices and Methods for Research
Switzerland AG: Springer Nature
Available at:
https://books.google.co.uk/books?hl=en&lr=&id=e87YDwAAQBAJ&oi=fnd&pg=PA323&dq=virtual+reality+standalone+pc&ots=eK3NjU77eX&sig=6H2Rhxy-2s9NLNjDtOpxVO5Cxjc#v=onepage&q=virtual%20reality%20standalone%20pc&f=false
(Accessed: 09/02/2020)

Unity (2021)
Choose the plan that is right for you
Available at:
https://store.unity.com/compare-plans
(Accessed: 29/04/2020)

Synthesis of Development

Andrew D Wilson (2006)
Robust Computer Vision-Based Detection of Pinching for One and Two-Handed Gesture Input
Available at:
https://dl.acm.org/doi/pdf/10.1145/1166253.1166292
(Accessed: 02/03/2021)

Evaluation

Jack Pottle (2021)
Virtual Nursing Simulation for Clinical Placements in the UK
https://oxfordmedicalsimulation.com/nursing/vr-simulation-clinical-placements/
(Accessed: 29/04/2021)

Mushtaq Hussain, Wenhao Zhu, Wu Zhang and Raza Abidi (2019)
Using machine learning to predict student difficulties from learning session data

Available at:
https://www.researchgate.net/publication/323082156_Using_machine_learning_to_predict_student_difficulties_from_learning_session_data
(Accessed: 27/04/2021)

Hamish Hector (2021)
Oculus Quest 2 sales figures prove VR has finally gone mainstream
Available at:
https://www.techradar.com/uk/news/oculus-quest-2-sales-figures-prove-vr-has-finally-gone-mainstream
(Accessed: 21/04/2021)

Dong Yoon Park (2019)
Microsoft Mixed Reality Toolkit 101: How to use Mixed Reality Toolkit Unity for common spatial interactions
Available at:
https://docs.microsoft.com/en-us/windows/mixed-reality/out-of-scope/mrtk-101
(Accessed 29/04/2021)

Appendices

Appendix 1. Terms of Reference

KV6003: Individual Computing Project

Project Terms of Reference

Andrew Nattress - W17009550 - Computer Science with Games Development

Developing an Immersive Educational Tool for Training Nurses and Reducing Misdiagnosis by Creating Engaging Learning Experiences.

Supervisor: Alan Godfrey, Secondary Marker: Hua Mao

Software Engineering Project

Contents

Project Background

Introduction

The opportunity to work on this project arose when the head of nurse training for the North East of England approached Boxmodel, the company that I work for, seeking a partnership to develop a new educational tool. The project aims to better engage learners and create a memorable experience that improves the learner's retention of details and information compared to more traditional teaching methods. The reason for this is that misdiagnosis currently costs the NHS huge amounts of money every year (£583 million in the past 3 years according to Health Business News). Improving the education of nurses should over time reduce the number of misdiagnoses that occur. To do this I intend to develop a virtual reality application that teaches users symptoms of conditions and helps them to retain the information they learn.

Before beginning my personal project, I have already developed a basic prototype that shows the potential of such a product. The prototype was never intended to be a final product and I will have to start fresh when developing the application for my project. This prototype has been helpful to ensure the product will be useful and that it will be a good investment for the client. It has been shown to about 10 CNTW trust employees, in positions such as "research and development executive" and ward nurses, and they have provided feedback that will help guide development and have led us to focus on providing good visuals and intuitive user experience design.

The concept of the project is that the user will be able to view symptoms on a 3d model and read information about a given physical condition, starting with but not limited to blisters and sores. This will enable them to see and understand the stages and progression of these conditions, hopefully leading them to better recognise symptoms when they are in an actual hospital environment with real patients. My hope for this project is that it will not only save the NHS money by stopping the excessive amount of misdiagnosis compensation paid but will also reduce the number of people's lives that are affected by this misdiagnosis. According to the NHS Resolution Report (2018) there were 155 successful claims for misdiagnosis leading to fatalities that year. If better education can stop just one of these not only will a life be saved but the NHS will save on average £174,000.

My Interest

I am excited to work on this project as I enjoy developing for virtual reality and the project involves researching effective methods of employing virtual reality for the use of education. It has the potential to solve a real problem in the medical field, especially in this country. The intended client for this project is Cumbria, Northumberland, Tyne and Wear NHS Trust (henceforth known as CNTW), however, if they do not decide to purchase the product Boxmodel has several other clients to pitch it to. However, we have had several meetings

with CNTW trust members and they were very impressed and interested in the prototype at the time.

This project will allow me to develop the existing Unity knowledge that I learnt during my placement year. This is my most valued skill at Boxmodel as I am the only Unity developer there so improving my skills and having greater proof of their usefulness in the company will be very helpful to my career. Also, adding this project to Boxmodel's portfolio and proving how valuable our virtual reality software can be will hopefully increase demand for similar projects from other existing clients.

According to Dr Kevin Cleary, the medical director of the National Patient Safety Agency, one of the primary reasons for misdiagnosis was lack of training as the Daily Mail reported in 2009. This proves that there is a need for better teaching methods for medical professionals and I believe that my software can provide this. In his study on the effectiveness of virtual reality in education, Akhan Akbulut noted that students saw "12% more successful results on average" when using their VR education system as opposed to the control group, this shows virtual reality has great potential to improve our education system.

Challenges

This project has many challenges involved, such as I will be developing for cutting edge hardware; specifically the Oculus Quest. I made this decision as it is the most portable and affordable solution on the market, which makes it ideal for teaching classes. However, this creates difficulties for development as virtual reality demands a lot of resources on standalone hardware, so the solution will have to balance performance with user experience.

There are other challenges involved in developing this project. I have never developed for virtual reality before this project, so I will be developing with a framework that is new to me. Also, creating good user experience design for virtual reality is challenging even for experienced developers as it is a new medium, which requires new methods of navigation and input that are still being actively researched. Christopher D. Wickens determined that having an intuitive user interface was paramount to having effective education in his software.

This project will require lots of research to identify effective methods of teaching in virtual reality. To create an effective system I have to learn how to teach effectively and how virtual reality can enhance that experience. I intend to use both primary and secondary research to develop the project. I will study academic papers, to inform my design decisions throughout the project and throughout development we at Boxmodel will also workshop ideas with potential users to test and improve the experience for learners, keeping their education as the top priority.

The intended product of this project is to be an educational platform in Virtual Reality that can give training nurses a memorable experience that can teach them to better recognise symptoms in patients and help them diagnose conditions such as blisters and sores. Nurses

will likely only be able to use software for short periods of time, this will be an important factor when designing the software as it will have to be easy to use so that users aren't required to spend lots of time discovering how to use the product. Also, navigation within the software will have to be quick and easy so that the users can focus on learning about the body instead of wasting time in menus. Finally, the information about conditions should be displayed in a way that is quick to process and users absorb and memorise it easily.

Proposed Work

Analysis

Before beginning this project I was able to develop a small prototype of the final product, which I then showed to several members of NHS staff, who tested and provided feedback to me. This will be very useful as it will allow me to make informed design decisions based on feedback from highly trained professionals at the top of the medical field. All of this feedback was verbal and informal, so it would be helpful to use the prototype to gather more useful and constructive feedback, however it may not be possible in the near future due to coronavirus restrictions. If this is the case I will rely on video demonstrations, which is suboptimal as it doesn't give the same experience, but I also have a flatmate who is training to be a nurse who has volunteered to provide feedback. This gives me 2 alternatives until the coronavirus restrictions are lifted, allowing me to proceed with my project.

Before I begin the project I will create a Literature Review. In order to better prepare for the project, I would like to further research effective teaching methods, particularly investigating existing successful implementations in virtual reality as I believe there is great potential in this new medium to change the way we learn. Also, since this experience is being built to educate I would like to ensure it does that as effectively as possible. Since virtual reality is a new medium, however, I will have to also research more generalised methods of teaching and evaluate whether they will be effective in virtual reality.

I can then use this literature review, combined with my preliminary research and the client's requirements, to update my plans for the project and create a detailed requirement specification. This will list everything that is required in the final product, allowing me to design the solution and plan my schedule of development for the next year.

Synthesis

My initial plans for this project are to produce a virtual reality training solution that is modular by design. What I mean by this is that the solution will allow new conditions to be implemented with relative ease, this will make the system more useful as it will be easy to expand. This will also be beneficial as if the system proves to be useful we will be able to scale it to teach more scenarios, however, it creates an interesting challenge as I will need to design a single data type that can represent any type of condition on any body part.

I also intend to design the system to be easy to use for beginners. For this reason, I have decided to utilize the Oculus Quest's hand tracking instead of controllers as most of the test users did not find the controllers very intuitive and remembering which buttons to press was confusing for them. This means abstracting the interactions to their core components to make them intuitive for hand tracking instead of controllers.

In order to prepare for development, I intend to produce a series of designs that help me to develop the system efficiently. I will produce UML, user case, interaction and state machine diagrams, which will help me to prepare for the user's interactions with the software and make sure the solution meets the design requirements.

Figure 1. Screenshot of the prototype application I developed to evaluate the effectiveness of the client proposal. It shows a foot with a condition named Erythema and allows the user to read information and step through the different stages.

The solution will display a given condition on a model part of the body as shown in Figure 1. It will also provide the user with information about the condition that should help them diagnose it correctly. I will explore different ways to convey this information to the users, that teach it effectively and are also inclusive to all students.

The desired product of my project is a virtual reality application that uses 3D models to create an effective teaching resource to educate users to better recognise symptoms and identify conditions in patients. The application should be quick to use and deliver a high-quality immersive experience in a short amount of time. The 3D model will display visual symptoms of varying stages and allow the user to step through these stages to help them recognise the category of a condition when they're faced with it in a real environment and determine the risks and severity of the condition.

Where the prototype was limited in scope to just a foot model, the final solution will have the option of many different body parts and it will be easy for me to add new conditions as the

client requires. While we will have to buy new textures for the 3D models, everything else will be data input that if needed could be passed to the client in future versions.

Review and Evaluation

Upon completion of the project, I will evaluate the solution I have developed and the processes I used to develop it. I will review my work, comparing the resultant product with my initial design specification and evaluating its fitness for the intended purpose. This section will help me to learn from the project, as I will be identifying the difficulties that I encountered and the areas of success.

Since the project is for a Boxmodel client the solution will be supported after I finish University, this makes the feedback I gather more useful as I will have time to incorporate improvements and, as a result, produce a better product for the client.

I will also focus part of my review on the designs I created for the project, discussing how they helped the project and the impact they had on development. Good designs should help the project by inspiring new ideas and identifying problems you would have in development before you encounter them, allowing you to work around them and save you time.

I will also reflect on the changes to the initial design during the development, I will review my reasons for these changes and evaluate whether they were a good decision in the final product and why my initial designs didn't plan for this. This is an important part of the project as it will help me to improve my design stage in future projects. Finally, I will discuss improvements and changes I would like to make going forward and other opportunities the project has for continued development.

Technology

I have decided to use Unity to develop the solution as it is currently regarded as the best engine for virtual reality development and I have a lot of experience developing in the engine. I also plan to use the Oculus Integration plugin as it provides a good high-level implementation of Oculus hardware, although the only experience I have with it is developing the initial prototype, it will be a good learning experience for me and should help produce a high-quality solution. Good Unity development relies heavily on inheritance and utilising object-oriented programming to create CPU efficient programs that meet the targeted 13ms frame requirement of the Oculus Quest on its limited mobile hardware. Creating a visually impressive result while keeping performance a priority is a challenge on the Quest, but I intend to achieve this as the quality of the visual result will be important to the overall quality of the product.

While this project is not a game, Unity is a popular 3d game engine, making the project very relevant to my course, Computer Science with Game Development. I decided to pursue a non-game related project after enjoying my placement so much and being offered the

opportunity to continue there not just part-time over my final year but after I graduate. This project will allow me to grow my game development skills while maintaining my connection to my current job, meaning that if I choose to transition into game development I will have evidence that I am capable of developing large projects in the engine.

Aims of the Project

- To investigate effective methods of teaching, such as Kolb's experiential learning cycle, and improve learning with the use of immersive technologies by implementing effective teaching methods.
- To develop an educational tool, that can help to reduce cases of misdiagnosis in the NHS.

Objectives

- Identify good teaching methods and explain how I will implement them in my project in my literature review. This information will be useful when designing the system and I will have evidence to present to the client for my decisions.
- Develop a requirement specification based upon my literature review, research gathered from my prototype demonstrations and my past interviews with CNTW employees.
- Learn how to better design software so that it improves the result of the product. I also plan on learning good UX design for virtual reality. While this area is still being researched, I aim to enable the user to pick up and use the application easily.
- Develop a virtual reality application that meets the requirements of the specification I wrote.
- Produce chapters of the project report as I complete objectives, this will allow me to explain how and why I make decisions going forward.
- Review the solution I developed by providing users who are training to be nurses access to the application in order to evaluate its effectiveness and receive feedback.
- Analyse the results from my project testing, discuss how this affected my designs and how the results compare to the findings of my initial research.
- Complete an evaluation of the product and its development process. I will discuss the successfulness of my implementation, based on the test feedback and the response of the client. I will also evaluate the development and my effectiveness in designing the software and implementing the needs of the client.

Skills

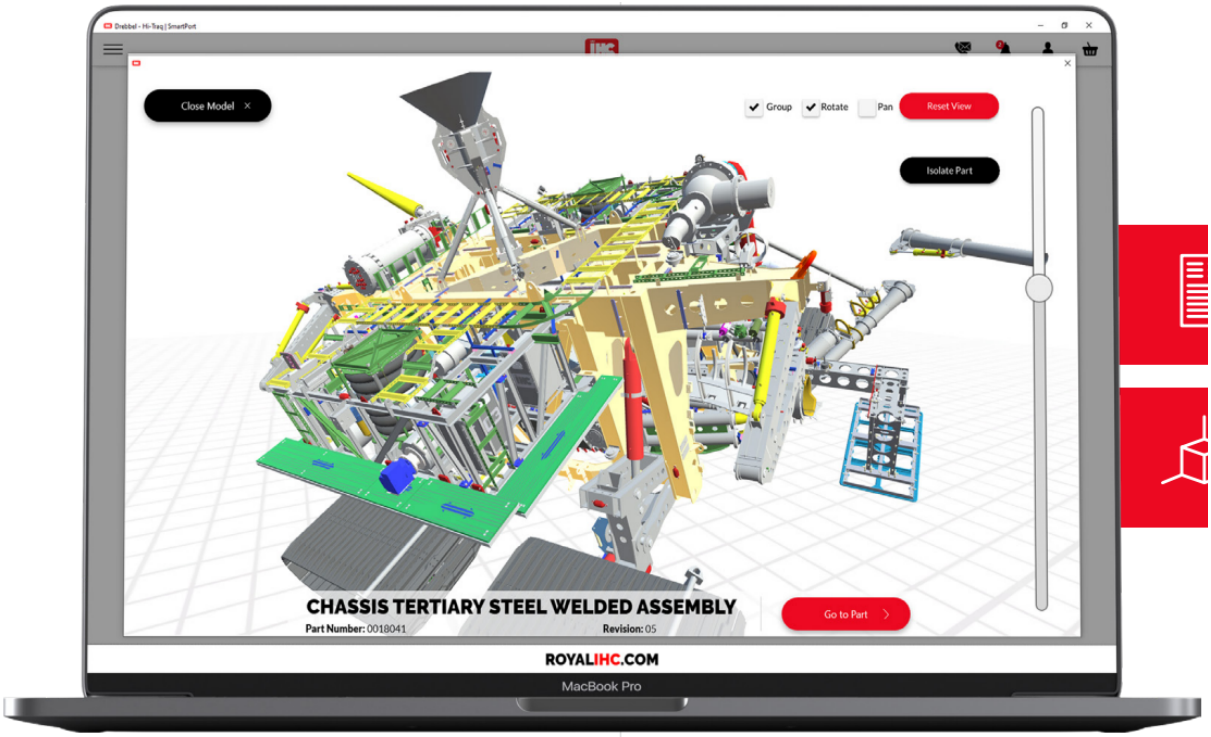


Figure 2. Demo of the interactive 3D viewer software I developed for a Boxmodel client.

I have proved my Unity development skills in a past project at Boxmodel. I developed a real-time 3D viewer to view complex 3D models that the client designs and creates. This application targeted mobile tablets and improved Boxmodel's relationship with the client by achieving something they couldn't previously do. The project was large and I was the sole developer of the 3D viewer aspect of the project. I overcame several challenges for the project but the largest was optimization for use on tablets. I will be able to use some of the techniques I learnt while researching for this project to optimise my virtual reality project for a standalone headset.

While I have experience with Unity, I have never developed a full virtual reality project before so I will be required to learn new skills to use Unity's virtual reality frameworks. I also will need to learn to deliver a polished virtual reality experience that provides feedback to the user and is pleasant to use.

I have good time management skills, I have proved this working on sprints in my time on placement. I will be given several tasks and required to complete them by the end of the week. This required time management skills when COVID-19 made us work from home and I began working flexible shifts.

An area in which I would like to improve my skills over the duration of the project is planning and design. While I have a small amount of experience designing projects and estimating duration, I am aware that it is an area I should improve to increase my value as a software developer. In order to improve, I will allow extra time in my plan for design and I will review my past University assignment feedback for improvements.

I'd also like to improve my research skills as it is essential for software developers to be able to learn new subjects when needed due to the rapidly changing technology industry. Currently, I am good at learning new skills, but in-depth research has never been a priority for me, which is why when learning completely new subjects such as machine learning I sometimes struggle.

Bibliography

Graysons (2016) What Are Really The Top Misdiagnosed Conditions In NHS Hospitals In 2014/15?
Available at:
https://www.graysons.co.uk/advice/the-top-misdiagnosed-conditions-in-nhs-hospitals/
(Accessed: 12/10/2020).

NHS Resolution (2018) NHS Resolution Freedom of Information F/3216
Available at:
https://resolution.nhs.uk/wp-content/uploads/2018/09/FOI_3216_Misdiagnosis.pdf
(Accessed: 14/10/2020)

Daily Mail (2009) One in six patients 'wrongly diagnosed by NHS doctors'
Available at:
https://www.dailymail.co.uk/health/article-1215243/One-patients-wrongly-diagnosed-NHS-doctors.html
(Accessed: 14/10/2020)

Akhan Akbulut (2018) On the effectiveness of virtual reality in the education of software engineering
Available at:
https://doi.org/10.1002/cae.21935
(Accessed: 15/10/2020)

C.D. Wickens (1992) Virtual reality and education
Available at:
https://ieeexplore.ieee.org/abstract/document/271688
(Accessed: 15/10/2020)

Resources

I will require an Oculus Quest to test my project on, I already own one myself, however, if something were to happen to it I have access to 2 more through my employer. I will also require a computer to develop the project, I have my own personal computer and a laptop supplied by my employer. I back up all of my work on GitHub, so if anything fails with either of the machines I have a backup I can continue development with. I also have a Unity Pro membership, this allows me to develop Unity products for commercial use for any company. I also have a company licenced Visual Studio account for developing for commercial use.

Structure and Contents of Project Report

Report structure:

- Abstract: A short introduction to the background of my project.
- Introduction: In the introduction to the report, I will discuss the reasons I chose to pursue this project, what I wish to achieve and learn completing the project and I will describe the intended final product I aim to deliver.
- Analysis of Research: In this section I will evaluate the findings of my research, discussing both my primary and secondary research, the methods used to obtain it and it's relevance to my project. I will also discuss the differences between my solution, similar existing platforms and current methods of teaching in the medical industry.
- Analysis of Requirements: The product requirement specification will be defined by the information retrieved from the literature review, my other research and the client's needs. In this section, I will review my decisions, the reasons for them and the compromises I made. This section will also review the decisions I make for the use of software and hardware to deliver the project.
- Synthesis of Designs: I will discuss the general design of the system I plan to make, discussing the reasons I came to specific design decisions. I will discuss my methods of design, and the reasoning behind my design decisions.
- Synthesis of Implementation: This chapter will discuss the development of the solution, reviewing its suitability for purpose and the development tools I used and extended. I will also examine the problems I encountered and areas for improvement.
- Synthesis of Testing: In this chapter, I will examine my testing process and the results gathered from it. I will provide justifications for the types of testing I used and evaluate their effectiveness. I will also reflect on how the results of my testing would change

the project if I could start over and if it had any effect on the project I deliver to the client.

- ➢ Evaluation of the Product: In this chapter, I will discuss the strengths and weaknesses of the final product. I will assess if the solution meets the initial requirements of my research and justify any deviations or changes I have made to the system, providing evidence to support my evaluation.
- ➢ Evaluation of the Project Process: I will review everything I have learnt over the duration of the project in this chapter. I will evaluate my performance on the project, what I did well and how I would do better in the future. I will also reflect on the process I planned to follow; whether it was helpful to the project and how closely I followed it.
- ➢ Conclusion: This chapter will summarize the main objectives of the project, review its success and I will present my own opinions based on my findings during the project. I will consider further work that might benefit the project and how the project has affected my skills and views of software development.

Appendices:

- Requirement Specification
- Design Documentation
- Source Code GitHub Repository
- Test Plans and Results

Marking Scheme:

Software Development Project

- Report - 40%
- Product - 50%
- Viva - 10%
- Total - 100%

Report

- Abstract & Introduction - 5%
- Analysis - 30%
- Synthesis - 30%
- Evaluation & Conclusions - 30%
- Presentation - 5%
- Total - 100%

Product

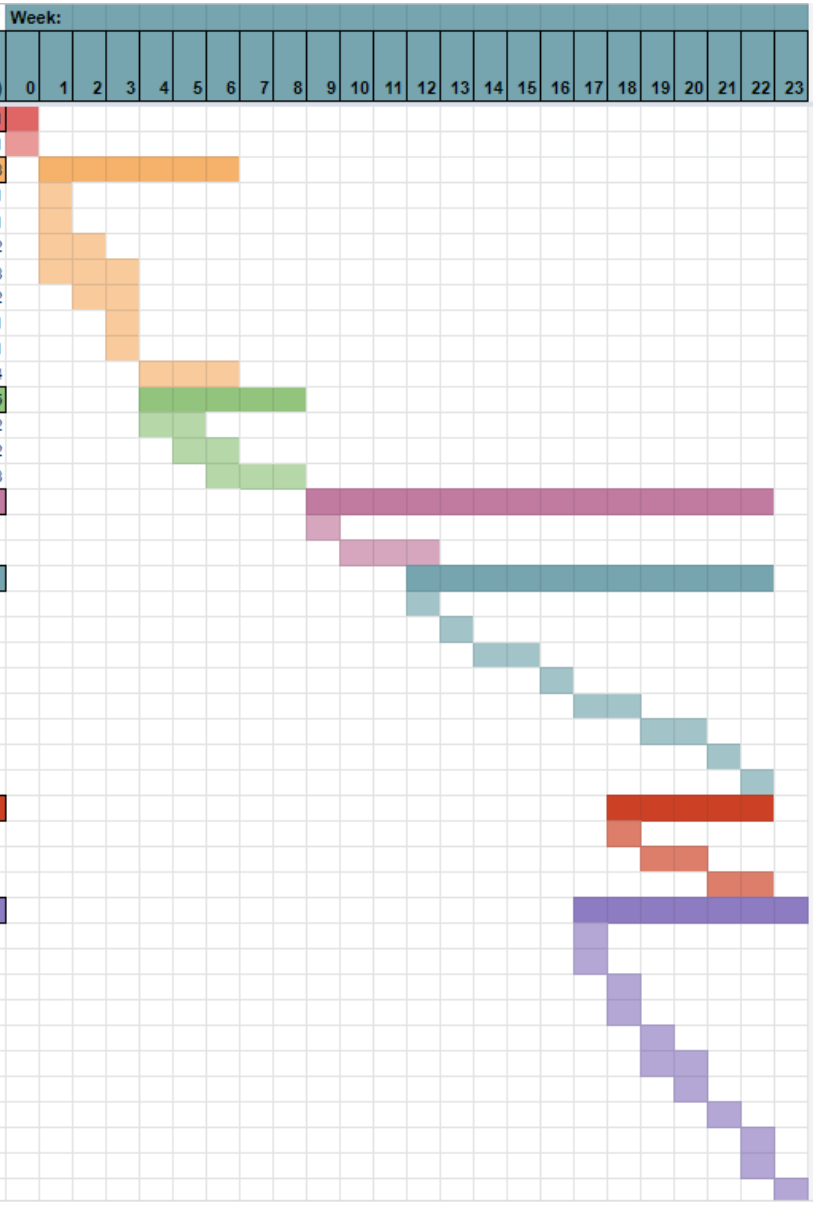
- Fitness for Purpose - 40%
- Build Quality - 60%
- Total - 100%

Viva

- Presentation / Demonstration - 50%
- Discussion - 50%
- Total - 100%

Project Plan

Task	Planned Start	Planned Length (Weeks)	Actual Start	Actual Length (Weeks)
Submit PID	0	1	0	1
Find Supervisor	0	1	0	1
Submit TOR	1	3	1	3
Title Project	1	1	1	1
Background Of Project	1	1	1	1
Proposed Work	1	2	1	2
Bibliography	1	3	1	3
Aims of Project, Objectives, Skills, Resources	2	2	2	2
Structure and Contents of Report	3	1	3	1
Project Plan	3	1	3	1
Feedback and Changes	4	3	4	4
Literature Review	4	5	4	5
Find relevant literature	4	2	4	2
Read literature and produce notes	5	2	5	2
Write Literature Review	6	3	6	3
Design Platform	9	14		
Produce requirement specification	9	1		
Produce UML, User Case, Interaction and State Machine Diagrams	10	3		
Develop Platform	12	10		
Setup Hand tracking	12	1		
Setup Room for learning with Hospital assets	13	1		
Create condition information storage framework	14	2		
Add foot 3D model, allow in edtior switching of stages	16	1		
Create UI capable of switching conditions and stages	17	2		
Develop editor testing tools and new condition creation tools	19	2		
Develop efficient shaders	21	1		
Response to Testing	22	1		
Test Application	18	5		
Organise Testing	18	1		
User Testing, interviews and survey	19	2		
Review Testing	21	2		
Write Report	17	7		
Write abstract	17	1		
Write Introduction	17	1		
Write Analysis of Research	18	1		
Write Analysis of Requirements	18	1		
Write Synthesis of Design	19	1		
Write Synthesis of Implementation	19	2		
Write Synthesis of Testing	20	1		
Write Evaluation of the Product	21	1		
Write Evaluation of the Project Process	22	1		
Write Conlusion	22	1		
Review Work	23	1		



Risk Assessment

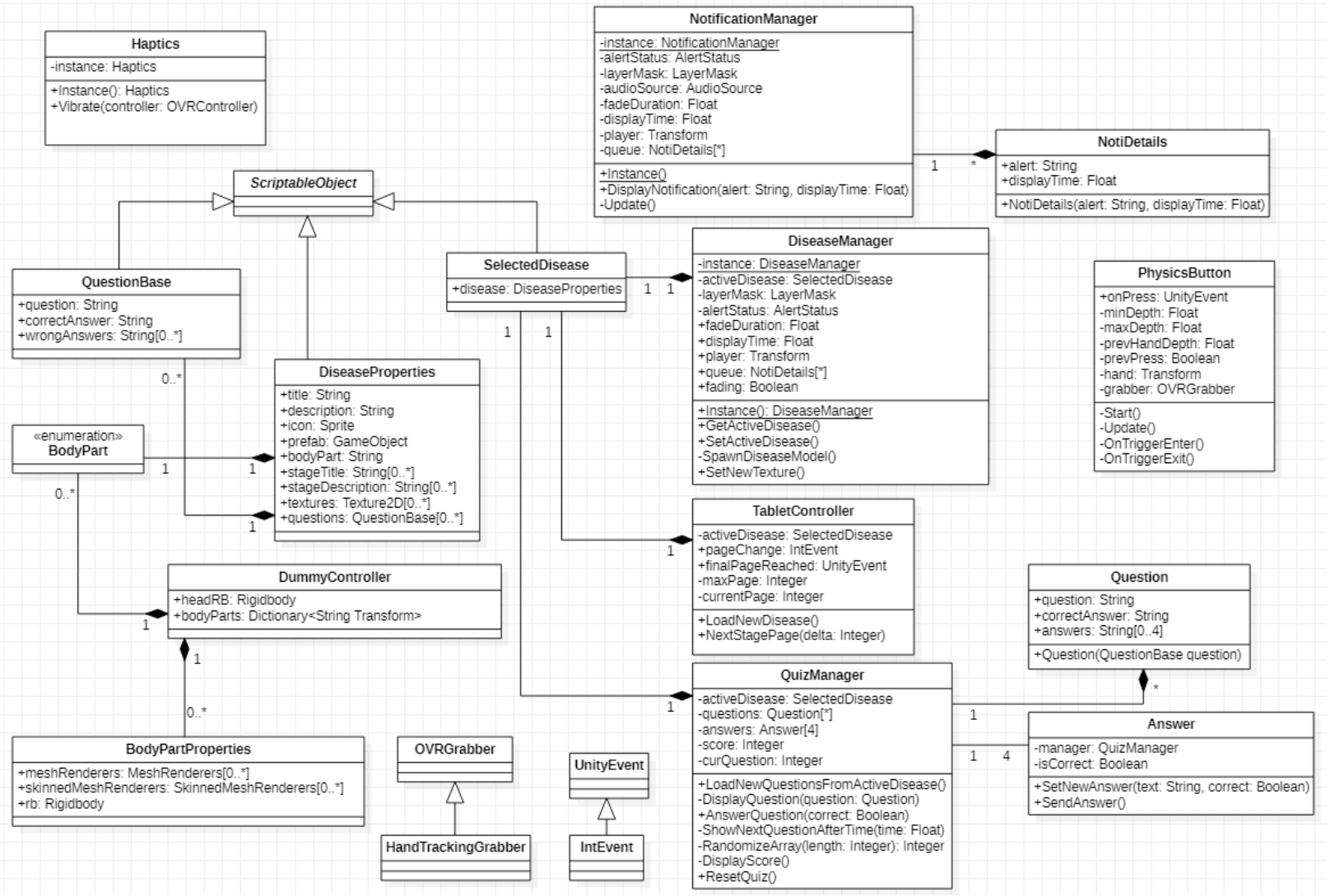
Description	Severity	Likelihood	Action
Unable to source new 3D models	Low	Medium	Develop the product to be able to have new models added at any time. The foot model will be enough to showcase the software.
Testers using virtual reality headset could injure themselves.	Medium	Low	Clear virtual reality playspace from obstructions. Monitor users in virtual reality to ensure nothing enters their space.

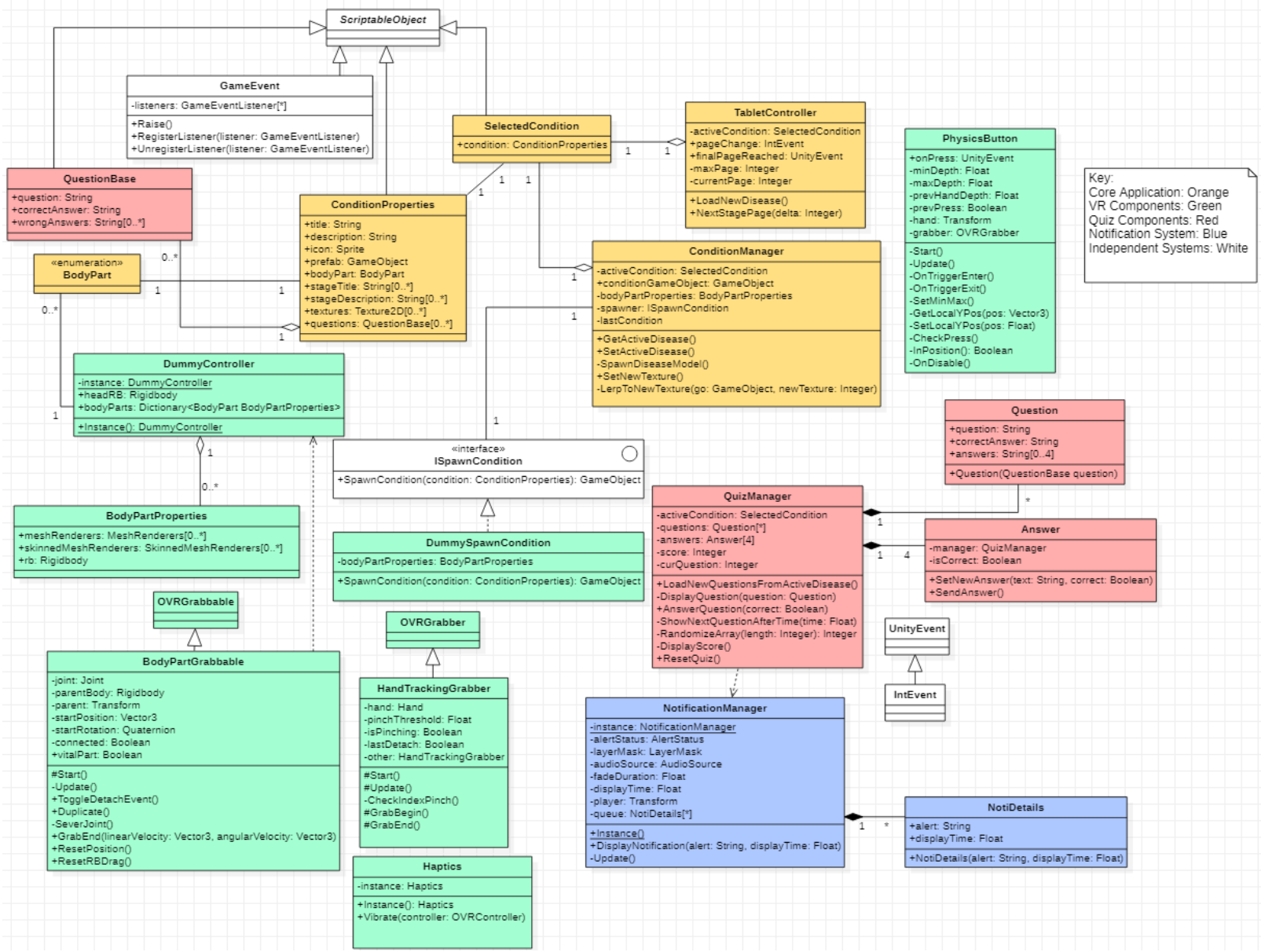
Appendix 2. Requirements Specification

The final product is expected to meet the following requirements:

1. Developers must be easily able to add new conditions and edit current ones.
2. The system should be easy for users to pick up and learn.
3. The solution must provide a realistic experience of each condition's expected symptoms
4. The solution should encourage users learning using research backed teaching methods.
5. The virtual environment should make users feel comfortable.
6. The system should include the option of hand tracking to reduce the technical skill needed to use the application.
7. The application should primarily be developed for the Oculus Quest platform to allow for portable use cases such as university visits.
8. The application should run at 72 frames per second on the original Quest and 90 frames per second on the Quest 2 to reduce vr sickness.
9. If possible support for other headsets such as the Oculus Rift and Valve Index would be beneficial.
10. The system must be well designed with low coupling and high cohesion, allowing for multiple platform design with relative ease.

Appendix 3. Nearly final and Final Draft Class Diagram




ScriptableObject
GameEvent
-listeners: GameEventListener[*]
+Raise()
+RegisterListener(listener: GameEventListener)
+UnregisterListener(listener: GameEventListener)
SelectedCondition
+condition: ConditionProperties
TabletController
-activeCondition: SelectedCondition
+pageChange: IntEvent
+finalPageReached: UnityEvent
-maxPage: Integer
-currentPage: Integer
+LoadNewDisease()
+NextStagePage(delta: Integer)
PhysicsButton
+onPress: UnityEvent
-minDepth: Float
-maxDepth: Float
-prevHandDepth: Float
-prevPress: Boolean
-hand: Transform
-grabber: OVRGrabber
-Start()
-Update()
-OnTriggerEnter()
-OnTriggerExit()
-SetMinMax()
-GetLocalYPos(pos: Vector3)
-SetLocalYPos(pos: Float)
-CheckPress()
-InPosition(): Boolean
-OnDisable()
Key:
Core Application: Orange
VR Components: Green
Quiz Components: Red
Notification System: Blue
Independent Systems: White
QuestionBase
+question: String
+correctAnswer: String
+wrongAnswers: String[0..*]
«enumeration»
BodyPart
ConditionProperties
+title: String
+description: String
+icon: Sprite
+prefab: GameObject
+bodyPart: BodyPart
+stageTitle: String[0..*]
+stageDescription: String[0..*]
+textures: Texture2D[0..*]
+questions: QuestionBase[0..*]
ConditionManager
-activeCondition: SelectedCondition
+conditionGameObject: GameObject
-bodyPartProperties: BodyPartProperties
-spawner: ISpawnCondition
-lastCondition
+GetActiveDisease()
+SetActiveDisease()
-SpawnDiseaseModel()
+SetNewTexture()
-LerpToNewTexture(go: GameObject, newTexture: Integer)
DummyController
-instance: DummyController
+headRB: Rigidbody
+bodyParts: Dictionary<BodyPart BodyPartProperties>
+Instance(): DummyController
«interface»
ISpawnCondition
+SpawnCondition(condition: ConditionProperties): GameObject
Question
+question: String
+correctAnswer: String
+answers: String[0..4]
+Question(QuestionBase question)
BodyPartProperties
+meshRenderers: MeshRenderers[0..*]
+skinnedMeshRenderers: SkinnedMeshRenderers[0..*]
+rb: Rigidbody
DummySpawnCondition
-bodyPartProperties: BodyPartProperties
+SpawnCondition(condition: ConditionProperties): GameObject
QuizManager
-activeCondition: SelectedCondition
-questions: Question[*]
-answers: Answer[4]
-score: Integer
-curQuestion: Integer
+LoadNewQuestionsFromActiveDisease()
-DisplayQuestion(question: Question)
+AnswerQuestion(correct: Boolean)
-ShowNextQuestionAfterTime(time: Float)
-RandomizeArray(length: Integer): Integer
-DisplayScore()
+ResetQuiz()
Answer
-manager: QuizManager
-isCorrect: Boolean
+SetNewAnswer(text: String, correct: Boolean)
+SendAnswer()
OVRGrabbable
OVRGrabber
UnityEvent
IntEvent
BodyPartGrabbable
-joint: Joint
-parentBody: Rigidbody
-parent: Transform
-startPosition: Vector3
-startRotation: Quaternion
-connected: Boolean
+vitalPart: Boolean
#Start()
-Update()
+ToggleDetachEvent()
+Duplicate()
-SeverJoint()
+GrabEnd(linearVelocity: Vector3, angularVelocity: Vector3)
+ResetPosition()
+ResetRBDrag()
HandTrackingGrabber
-hand: Hand
-pinchThreshold: Float
-isPinching: Boolean
-lastDetach: Boolean
-other: HandTrackingGrabber
#Start()
#Update()
-CheckIndexPinch()
#GrabBegin()
#GrabEnd()
NotificationManager
-instance: NotificationManager
-alertStatus: AlertStatus
-layerMask: LayerMask
-audioSource: AudioSource
-fadeDuration: Float
-displayTime: Float
-player: Transform
-queue: NotiDetails[*]
+Instance()
+DisplayNotification(alert: String, displayTime: Float)
-Update()
NotiDetails
+alert: String
+displayTime: Float
+NotiDetails(alert: String, displayTime: Float)
Haptics
-instance: Haptics
+Instance(): Haptics
+Vibrate(controller: OVRController)

Appendix 4. Stakeholder Feedback

boxmodel

The product Andrew has developed began during his placement year when a client approached us with an idea for an augmented reality application that improves education in the medical field. The problem suggested that a high number of misdiagnosis claims within the National Health Services were caused by ineffective education.

Over the past year, Andrew has worked as a full-time employee at Boxmodel around his studies developing a virtual reality application that is designed to be used alongside regular teaching methods to improve the educational experience of medical courses for nursing students. The generalised nature of Andrew's system architecture allows us to pitch the application to many sectors in the medical industry and in other industries that could benefit from a virtual reality education experience. The application has been used in several pitches to clients and has made a positive impact each time, it has also provided us with potential clients to take the product further in the future. I am very happy with how the application has progressed as a result of Andrew's work.

While the original product pitch was for an augmented reality application, the client (a nursing educator on the board of the Cumbria, Northumberland, Tyne and Wear Trust) was extremely impressed with the early prototype of the virtual reality application that Andrew suggested as an option. This led to him suggesting it would be the better format for continued development due to the better immersion and potential for interaction.

Andrew's skill in designing Unity applications has allowed him to develop a solution capable of rapid adaptation to client requirements, the solution can be developed to whatever requirements they decide and change based upon their feedback. His approach to development is impressive as it allows lots of code reuse if the client wants the solution to be available on multiple platforms, aside from virtual reality applications we are pitching the possibility of augmented reality, mobile, desktop and web-based applications all from his single solution, requiring just an estimated 1-2 weeks worth of work to implement input systems for a new platform.

Appendix 5. Code

Condition Manager

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

public class ConditionManager : MonoBehaviour
{
    [HideInInspector] public GameObject conditionGameObject;
    ISpawnCondition spawner;
    string lastCondition = "";
    public SelectedCondition activeCondition;
    private void Awake()
    {
        spawner = GetComponent<ISpawnCondition>();
    }
    public ConditionProperties GetActiveCondition()
    {
        return activeCondition.condition;
    }
    public void SetActiveCondition()
    {
        if (activeCondition.condition.title == lastCondition)
        {
            activeCondition.condition = null;
            lastCondition = "";
            Destroy(conditionGameObject);
        }

        if (activeCondition.condition)
            lastCondition = activeCondition.condition.title;
        RefreshCondition();
    }
    public void RefreshCondition()
    {
        if (conditionGameObject)
            Destroy(conditionGameObject);
        conditionGameObject =
spawner.SpawnCondition(activeCondition.condition);
    }
    public void SetNewTexture(int i)
    {

conditionGameObject.GetComponentInChildren<Renderer>().material.mainTexture
= activeCondition.condition.textures[i];
    }
    public void LerpToTexture(int newTexture)
    {
        StartCoroutine(LerpToNewTexture(conditionGameObject, newTexture));
    }

    //maybe convert to static class and method probs just pass newTexture2D
    public IEnumerator LerpToNewTexture(GameObject go, int newTexture)
    {
        if (go != null)
```

```
        {
            Material mat = go.GetComponentInChildren<Renderer>().material;
            mat.SetTexture("_SecondaryTex",
activeCondition.condition.textures[newTexture]);

            float t = Time.time;
            while (Time.time - t <= 1)
            {
                mat.SetFloat("_LerpValue", Time.time - t);

                yield return null;
            }
            mat.SetFloat("_LerpValue", 0);
            mat.SetTexture("_MainTex",
activeCondition.condition.textures[newTexture]);
        }
    }
}
```

Condition Properties

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

[CreateAssetMenu(menuName = "Scriptables/Condition Property")]
public class ConditionProperties : ScriptableObject
{
    public string title;
    public Sprite icon;
    [TextArea(5, 12)]
    public string description;
    public ModelPrefab prefab;
    public BodyPart bodyPart;
    public string[] stageTitle;
    [TextArea(1, 12)]
    public string[] stageDescription;
    public Texture2D[] textures;
    public Texture2D[] binderTextures;

    public List<QuestionBase> questions;
}
```

Selected Condition

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
[CreateAssetMenu]
public class SelectedCondition : ScriptableObject
{
    public ConditionProperties condition;
    void Awake()
    {
        condition = null; //sets it to null as editor changes save;
    }
```

```
}
```

Physics Button

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

public class PhysicsButton : MonoBehaviour
{
    public UnityEvent onPress;
    [Header("Debug Values")]
    [SerializeField] float minDepth, maxDepth;
    float prevHandDepth;
    bool prevPress = false;
    Transform hand;
    OVRGrabber g;
    private void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("Hand"))
        {
            hand = other.transform;
            prevHandDepth = GetLocalYPos(hand.position);
            g = hand.GetComponentInParent<OVRGrabber>();
        }
    }
    private void OnTriggerExit(Collider other)
    {
        if (other.CompareTag("Hand"))
        {
            hand = null;
            prevHandDepth = 0;
            prevPress = false;
            SetLocalYPos(maxDepth);
        }
    }
    private void Start()
    {
        SetMinMax();
    }
    private void Update()
    {
        if (hand)
        {
            float newHandDepth = GetLocalYPos(hand.position);
            float handDifference = prevHandDepth - newHandDepth;
            prevHandDepth = newHandDepth;
            float newPos = transform.localPosition.y - handDifference;
            SetLocalYPos(newPos);

            CheckPress();
        }
    }
    void SetMinMax()
    {
```

```
        Collider buttonCollider = GetComponent<Collider>();
        float yBounds = buttonCollider.bounds.size.y;
        if (Mathf.Abs(Vector3.Dot(transform.up, Vector3.forward)) > 0.8)
            yBounds = buttonCollider.bounds.size.z;
        else if (Mathf.Abs(Vector3.Dot(transform.up, Vector3.right)) > 0.8)
            yBounds = buttonCollider.bounds.size.x;
        minDepth = transform.localPosition.y - (yBounds * 0.5f);
        maxDepth = transform.localPosition.y;
    }
    float GetLocalYPos(Vector3 pos)
    {
        Vector3 localPosition =
transform.parent.InverseTransformPoint(pos);
        return localPosition.y;
    }
    void SetLocalYPos(float pos)
    {
        Vector3 newPosition = transform.localPosition;
        newPosition.y = Mathf.Clamp(pos, minDepth, maxDepth);
        transform.localPosition = newPosition;
    }
    void CheckPress()
    {
        bool inPos = InPosition();
        if (inPos && !prevPress)
        {
            if (g && MyHaptics.instance)
                MyHaptics.instance.Vibrate(0.3f, 0.3f, 0.1f, !g.isLeftHand,
g.isLeftHand);
            onPress.Invoke();
        }
        prevPress = inPos;
    }
    [ContextMenu("Test Press")]
    public void TestPress()
    {
        onPress.Invoke();
    }
    private bool InPosition()
    {
        float inRange = Mathf.Clamp(transform.localPosition.y, minDepth,
minDepth + 0.01f);

        return transform.localPosition.y == inRange;
    }
    private void OnDisable()
    {
        hand = null;
        prevHandDepth = 0;
        prevPress = false;
        SetLocalYPos(maxDepth);
    }
}
```

Tablet Controller

```
using System.Collections;
```

```
using System.Collections.Generic;
using TMPro;
using UnityEngine;
using UnityEngine.Events;

[System.Serializable]
public class IntEvent : UnityEvent<int> { }
public class TabletController : MonoBehaviour
{
    public SelectedCondition activeCondition;
    [SerializeField] TextMeshProUGUI titleText;
    [SerializeField] TextMeshProUGUI descriptionText;
    [SerializeField] TextMeshProUGUI stageText;
    [SerializeField] GameObject backPage;
    [SerializeField] GameObject fwdPage;

    [SerializeField] IntEvent pageChanged = new IntEvent();
    public UnityEvent finalPageReached = new UnityEvent();
    int currentPage = 0;
    int maxPage;

    public void LoadNewCondition()
    {
        currentPage = 0;
        if (activeCondition.condition)
        {
            titleText.text = activeCondition.condition.title;
            descriptionText.text = activeCondition.condition.description +
"\n" + activeCondition.condition.stageDescription[currentPage];
            stageText.text =
activeCondition.condition.stageTitle[currentPage];


            maxPage = activeCondition.condition.stageTitle.Length;
            if (currentPage + 1 < maxPage)
                fwdPage.SetActive(true);
            else
                fwdPage.SetActive(false);
            backPage.SetActive(false);

            hasReachedFinalPage = false;
        }
        else
        {
            titleText.text = "No condition selected";
            descriptionText.text = "";
            stageText.text = "";
            fwdPage.SetActive(false);
            backPage.SetActive(false);
        }
    }
    bool hasReachedFinalPage = false;
    [ContextMenu("Next Stage")]
    public void NextStageTest()
    {
        NextStagePage(1);
    }
    [ContextMenu("Prev Stage")]
```

```
    public void PrevStageTest()
    {
        NextStagePage(-1);
    }
    public void NextStagePage(int delta)
    {
        if (currentPage + delta < maxPage && currentPage + delta >= 0)
        {
            currentPage += delta;
            StartCoroutine(SetDesc(activeCondition.condition.description +
"\n" + activeCondition.condition.stageDescription[currentPage]));

stageText.SetText(activeCondition.condition.stageTitle[currentPage]);
            if (currentPage + 1 < maxPage)
                fwdPage.SetActive(true);
            else
            {
                fwdPage.SetActive(false);

                if (!hasReachedFinalPage)
                    finalPageReached.Invoke();
                hasReachedFinalPage = true;
            }
            if (currentPage > 0)
                backPage.SetActive(true);
            else
                backPage.SetActive(false);

            pageChanged.Invoke(currentPage);
        }
    }
    IEnumerator SetDesc(string s)
    {
        yield return null;
        yield return null;
        descriptionText.SetText(s);
    }
}
```

Binder Controller

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

public class BinderController : MonoBehaviour
{
    public SelectedCondition activeCondition;
    public Animator anim;

    [SerializeField] BinderPage[] pages;

    [SerializeField] GameObject backPage;
    [SerializeField] GameObject fwdPage;
```

```
    [SerializeField] IntEvent pageChanged = new IntEvent();
    public UnityEvent finalPageReached = new UnityEvent();

    [SerializeField] int currentPage = 0;
    [SerializeField] int maxPage;
    [SerializeField] int basePages;
    bool hasReachedFinalPage = false;
    private void Start()
    {
        LoadNewCondition();
    }
    public void LoadNewCondition()
    {
        if (activeCondition.condition)
        {
            maxPage = basePages +
activeCondition.condition.binderTextures.Length - 1;

            fwdPage.SetActive(currentPage + 1 < maxPage);
            backPage.SetActive(currentPage > -1);

            for (int i = 0; i <
activeCondition.condition.binderTextures.Length; i++)
            {
                if (basePages + i < pages.Length)
                {
                    pages[basePages + i].gameObject.SetActive(true);
                    pages[basePages +
i].SetNewTexture(activeCondition.condition.binderTextures[i]);
                }
            }
            for (int i = maxPage + 1; i < pages.Length; i++)
                pages[i].gameObject.SetActive(false);
            NextStagePage(-currentPage, true); // return to 0;
            hasReachedFinalPage = false;
        }
        else
        {
            maxPage = basePages - 1;
            NextStagePage(-currentPage - 1, true);
            //hide all and close when active condition is null;
        }
    }
    //currently only works on 1 page
    public void NextStagePage(int delta, bool skipEvents = false)
    {
        if (currentPage + delta <= maxPage && currentPage + delta >= -1)
        {
            int lastPage = currentPage;
            currentPage += delta;
            if (currentPage >= maxPage)
            {
                fwdPage.SetActive(false);

                if (!hasReachedFinalPage && !skipEvents)
                    finalPageReached.Invoke();
                hasReachedFinalPage = true;
```

```
            }
            else
            {
                fwdPage.SetActive(true);
            }

            backPage.SetActive(currentPage > -1);


            if (currentPage != -1 && !skipEvents)
                pageChanged.Invoke(currentPage-basePages);

            int step = Mathf.Clamp(currentPage - lastPage, -1, 1);
            for (int i = lastPage; i != currentPage + (step < 0 ? -1 : 0);
i += step)
            {
                if (i == -1)
                {
                    //binder Open/Close
                    if (step > 0)
                        OpenBinder();
                    else
                        CloseBinder();

                }
                else
                    pages[i].ChangePage(step);
            }

        }
    }
    void OpenBinder()
    {
        // anim.SetFloat("Direction", 1);
        anim.Play("Open");// 0, 1.0f);
    }
    void CloseBinder()
    {
        //anim.SetFloat("Direction", -1);
        anim.Play("Close");//, 0, -Time.deltaTime);
    }

    [ContextMenu("Next Stage")]
    public void NextStageTest()
    {
        NextStagePage(1);
    }
    [ContextMenu("Prev Stage")]
    public void PrevStageTest()
    {
        NextStagePage(-1);
    }
}
```

Binder Page

```
using System.Collections;
```

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

public class BinderPage : MonoBehaviour
{
    public MeshRenderer meshRenderer;
    public MeshAnimator meshAnim;
    MaterialPropertyBlock propertyBlock;
    void Awake()
    {
        if (meshRenderer == null)
            meshRenderer = GetComponent<MeshRenderer>();
        if (meshAnim == null)
            meshAnim = GetComponent<MeshAnimator>();

        propertyBlock = new MaterialPropertyBlock();
    }

    public void SetNewTexture(Texture2D tex)
    {
        propertyBlock.SetTexture("_MainTex", tex);
        meshRenderer.SetPropertyBlock(propertyBlock, 0);
    }
    public void ChangePage(int d)
    {
        if (d > 0)
            StartCoroutine(meshAnim.ForwardPage());
        else if (d < 0)
            StartCoroutine(meshAnim.BackPage());
    }
}
```

ModelPrefab

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

public class ModelPrefab : MonoBehaviour
{
    public Renderer rend;
    public Collider collision;
}
```

ISpawnCondition

```
using UnityEngine;
public interface ISpawnCondition
{
    GameObject SpawnCondition(ConditionProperties condition);
}
```

Dummy Spawn Condition

```
using System.Collections;
```

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

[System.Serializable]
public class DummyConditionSpawner : MonoBehaviour, ISpawnCondition
{
    BodyPartProperties bodyPartProperties;
    public GameObject SpawnCondition(ConditionProperties condition)
    {
        if (bodyPartProperties)
        {
            foreach (SkinnedMeshRenderer smr in
bodyPartProperties.meshRenderers)
                smr.enabled = true;

            foreach (MeshRenderer mr in
bodyPartProperties.meshRenderersReg)
                mr.enabled = true;
        }
        GameObject conditionGameObject = null;
        if (condition)
        {

DummyController.Instance.bodyParts.TryGetValue(condition.bodyPart, out
bodyPartProperties);

            conditionGameObject = Instantiate(condition.prefab,
bodyPartProperties.transform).gameObject;
            conditionGameObject.transform.localPosition = Vector3.zero;
            conditionGameObject.transform.localRotation =
Quaternion.identity;

            foreach (SkinnedMeshRenderer smr in
bodyPartProperties.meshRenderers)
                smr.enabled = false;
            foreach (MeshRenderer mr in
bodyPartProperties.meshRenderersReg)
                mr.enabled = false;
        }
        return conditionGameObject;
    }
}
```

Rotator Spawn Condition

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

public class RotatorConditionSpawner : MonoBehaviour, ISpawnCondition
{
    public GameObject SpawnCondition(ConditionProperties condition)
    {

        ModelPrefab conditionGameObject = null;
        if (condition)
        {
```

```
            conditionGameObject = Instantiate(condition.prefab, transform);
            conditionGameObject.transform.localPosition = Vector3.zero;
            conditionGameObject.transform.localRotation =
Quaternion.identity;
            conditionGameObject.transform.localScale *= 18.0f;
            conditionGameObject.gameObject.layer = 10;
            Rigidbody rb =
conditionGameObject.gameObject.AddComponent<Rigidbody>();
            rb.isKinematic = true;

            conditionGameObject.collision.enabled = true;
            OVRGrabbable g =
conditionGameObject.gameObject.AddComponent<PartGrabbable>();

            g.grabPoints = new Collider[1] { conditionGameObject.collision
};

        }
        return conditionGameObject.gameObject;
    }
}
```

Rotator

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

public class Rotator : MonoBehaviour
{
    Vector3 angle = Vector3.zero;
    [SerializeField] float speed;
    void Update()
    {
        if (grabbable && !grabbable.isGrabbed && grabbable.transform.parent
!= transform)
        {
            grabbable.transform.parent = transform;
        }
        angle.y = Mathf.Lerp(0, 360, (Time.time * speed) % 1);
        transform.eulerAngles = angle;
    }
    OVRGrabbable grabbable;
    private void OnTriggerEnter(Collider other)
    {
        grabbable = other.GetComponentInParent<OVRGrabbable>();

    }
    private void OnTriggerExit(Collider other)
    {
        OVRGrabbable g = other.GetComponentInParent<OVRGrabbable>();
        if (g && grabbable == g)
            grabbable = null;
    }
}
```

Part Grabbable

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

public class PartGrabbable : OVRGrabbable
{
    public bool isCopy = false;
    [ContextMenu("Duplication")]
    public virtual GameObject DuplicateHeldObject()
    {
        PartGrabbable d = Instantiate(this);
        d.isCopy = true;

        d.transform.position = transform.position;
        d.transform.rotation = transform.rotation;


        ModelPrefab p = d.gameObject.GetComponentInChildren<ModelPrefab>();
        if (p)
        {
            Renderer r = p.rend;
            Material mat = Instantiate(r.material);
            mat.name = "Instance " + Random.Range(0, 1000).ToString();
            r.material = mat;
        }
        return d.gameObject;
    }

    public override void GrabBegin(OVRGrabber hand, Collider grabPoint)
    {
        base.GrabBegin(hand, grabPoint);
        GrabbableControlsController.Instance.OnGrab(this);
    }
    public override void GrabEnd(Vector3 linearVelocity, Vector3
angularVelocity)
    {
        GrabbableControlsController.Instance.OnRelease(this);
        base.GrabEnd(linearVelocity, angularVelocity);
    }

    public void DestroyCopy()
    {
        Destroy(gameObject);
    }
}
```

Body Part Grabbable

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

public class BodyPartGrabbable : PartGrabbable
{
    [SerializeField] Joint joint;
```

```
    [SerializeField] Rigidbody parentedBody;
    Transform parent;
    Vector3 startPos;
    Quaternion startRot;
    public bool connected = true;
    public bool vitalPart = true;


    protected override void Start()
    {
        base.Start();

        parent = transform.parent;
        startPos = transform.localPosition;
        startRot = transform.localRotation;
        if (!joint)
            joint = GetComponent<Joint>();
    }
    private void Update()
    {
        if (isGrabbed)
        {
            if (grabbedBy)
            {
                if (parentedBody.angularVelocity.magnitude > 30 &&
connected)
                {
                    grabbedBy.ForceRelease(this);
                }
                if (OVRInput.GetActiveController() ==
OVRInput.Controller.Touch)
                {
                    //Detcatch
                    if ((grabbedBy.isLeftHand &&
OVRInput.GetDown(OVRInput.Button.Four)) || (!grabbedBy.isLeftHand &&
OVRInput.GetDown(OVRInput.Button.Two)))
                    {
                        ToggleDetachEvent();
                    }
                    //Duplicate
                    else if (!connected && (
                        (grabbedBy.isLeftHand &&
OVRInput.GetDown(OVRInput.Button.PrimaryIndexTrigger)) || //was three
                        (!grabbedBy.isLeftHand &&
OVRInput.GetDown(OVRInput.Button.One))))
                    {
                        DuplicateHeldObject();
                    }
                }
            }
        }
    }
    [ContextMenu("Toggle Detach")]
    public void ToggleDetachEvent()
    {
        if (joint && vitalPart)
        {
            if (connected)
```

```
            {
                SeverJoint();
            }
            else
            {
                ResetPosition();
            }
        }
        else if (!vitalPart)
        {
            Destroy(gameObject);
        }
    }

    public void SeverJoint()
    {
        joint.connectedBody = null;
        connected = false;
        Rigidbody rb = gameObject.GetComponent<Rigidbody>();
        rb.isKinematic = true;
    }

    public override void GrabEnd(Vector3 linearVelocity, Vector3
angularVelocity)
    {
        GrabbableControlsController.Instance.OnRelease(this);
        if (connected)
            base.GrabEnd(linearVelocity, angularVelocity);
        else
        {
            Rigidbody rb = gameObject.GetComponent<Rigidbody>();
            rb.isKinematic = true;
            rb.velocity = linearVelocity;
            rb.angularVelocity = angularVelocity;
            m_grabbedBy = null;
            m_grabbedCollider = null;
        }

    }
    public override GameObject DuplicateHeldObject()
    {
        GameObject dupe = base.DuplicateHeldObject();
        dupe.GetComponent<BodyPartGrabbable>().vitalPart = false;
        return dupe;
    }

    public void ResetPosition()
    {
        DummyController.Instance.SetRBDrag(3000);
        transform.parent = parent;
        transform.localPosition = startPos;
        transform.localRotation = startRot;
        joint.connectedBody = parentedBody;
        connected = true;
        Rigidbody rb = gameObject.GetComponent<Rigidbody>();
        rb.isKinematic = false;
        m_grabbedBy = null;
        m_grabbedCollider = null;
```

```
        StartCoroutine(ResetRBDrag());
    }
    IEnumerator ResetRBDrag()
    {
        yield return new WaitForSeconds(0.3f);
        DummyController.Instance.SetRBDrag(0.2f);
    }
}
```

Dummy Controller

```
using RotaryHeart.Lib.SerializableDictionary;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[System.Serializable]
public class StringTransformDictionary :
SerializableDictionaryBase<BodyPart, BodyPartProperties> { }
public class DummyController : MonoBehaviour
{
    #region Singleton
    private static DummyController instance = null;
    // Game Instance Singleton
    public static DummyController Instance
    {
        get
        {
            return instance;
        }
    }
    private void Awake()
    {
        // if the singleton hasn't been initialized yet
        if (instance != null && instance != this)
        {
            Destroy(this.gameObject);
        }

        instance = this;
    }
    #endregion
    public StringTransformDictionary bodyParts = new
StringTransformDictionary();
    public Rigidbody head;

    public void SetRBDrag(float v)
    {
        foreach (BodyPartProperties bpp in bodyParts.Values)
        {
            bpp.rb.drag = v;
            bpp.rb.angularDrag = v;
        }
    }
}
```

Body Part Enumerable

```
public enum BodyPart
{
    Chest,
    LeftFoot,
    RightFoot,
    LeftForearm,
    RightForearm,
    LeftHand,
    RightHand,
    Head,
    Pelvis,
    LeftShin,
    RightShin,
    LeftThigh,
    RightThigh,
    LeftUpperArm,
    RightUpperArm
}
```

Body Part Properties

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

public class BodyPartProperties : MonoBehaviour
{
    public SkinnedMeshRenderer[] meshRenderers;
    public MeshRenderer[] meshRenderersReg;
    public Rigidbody rb;
}
```

My Haptics

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

public class MyHaptics : MonoBehaviour
{
    public static MyHaptics instance;
    private void Awake()
    {
        if (instance != null && instance != this)
            Destroy(instance);
        instance = this;
    }
    public void Vibrate(float frequency, float amplitude, float duration,
bool rightHand, bool leftHand)
    {
        StartCoroutine(Haptics(frequency, amplitude, duration, rightHand,
leftHand));
    }
    IEnumerator Haptics(float frequency, float amplitude, float duration,
bool rightHand, bool leftHand)
```

```
    {
        if (rightHand) OVRInput.SetControllerVibration(frequency,
amplitude, OVRInput.Controller.RTouch);
        if (leftHand) OVRInput.SetControllerVibration(frequency, amplitude,
OVRInput.Controller.LTouch);

        yield return new WaitForSeconds(duration);

        if (rightHand) OVRInput.SetControllerVibration(0, 0,
OVRInput.Controller.RTouch);
        if (leftHand) OVRInput.SetControllerVibration(0, 0,
OVRInput.Controller.LTouch);
    }
}
```

Quiz Manager

```
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using TMPro;
using UnityEngine;

public class QuizManager : MonoBehaviour
{
    public SelectedCondition activeCondition;

    public GameObject questionScreen;
    public GameObject scoreScreen;
    public GameObject correctScreen;
    public GameObject wrongScreen;
    public TextMeshProUGUI wrongScreenAnswer;
    [SerializeField] TextMeshProUGUI qText;
    [SerializeField] TextMeshProUGUI scoreText;
    [SerializeField] Answer[] aTexts;
    [SerializeField] List<Question> qs = new List<Question>();

    int score = 0;
    int curQuestion = 0;

    public void LoadNewQuestionsFromActiveCondition()
    {
        if (activeCondition.condition == null)
            return;

        List<QuestionBase> newQs = new
List<QuestionBase>(activeCondition.condition.questions);
        int c = newQs.Count;
        int newQuestionCount = 0;
        for (int i = 0; i < c; i++)
        {

            int x = Random.Range(0, newQs.Count);
            Question q = Question.ConvertBase(newQs[x]);
            if (!qs.Contains(q))
            {
                qs.Add(q);
```

```
                ++newQuestionCount;
            }
            newQs.RemoveAt(x);

        }
        if (newQuestionCount > 0)
            NotificationManager.instance.DisplayNotification($"Added
{newQuestionCount} Question{(newQuestionCount > 1 ? "s" : "")} to the
quiz");


        if (curQuestion < qs.Count)
            DisplayQuestion(qs[curQuestion]);

    }
    void DisplayQuestion(Question q)
    {
        questionScreen.SetActive(true);
        qText.SetText(q.question);
        int answerCount = q.answers.Count;

        int[] i = RandomArray(answerCount + 1);//fixes boundary
        for (int c = 0; c < i.Length; c++)
        {
            if (i[c] == answerCount)
                aTexts[c].SetNewAnswer(q.correctAnswer, true);
            else
                aTexts[c].SetNewAnswer(q.answers[i[c]], false);
        }
    }
    public void AnswerQuestion(bool correct)
    {
        questionScreen.SetActive(false);
        if (correct)
        {
            score++;
            correctScreen.SetActive(true);
        }
        else
        {
            wrongScreen.SetActive(true);
            wrongScreenAnswer.SetText(qs[curQuestion].correctAnswer);
        }
        curQuestion++;
        StartCoroutine(ShowNextQuestionAfterTime(1.5f));
    }
    IEnumerator ShowNextQuestionAfterTime(float time)
    {
        yield return new WaitForSeconds(time);
        wrongScreen.SetActive(false);
        correctScreen.SetActive(false);
        if (curQuestion < qs.Count)
            DisplayQuestion(qs[curQuestion]);
        else
            DisplayScore();

    }
    int[] RandomArray(int maxRand)
```

```
    {
        List<int> list = new List<int>();
        for (int i = 0; i < maxRand; i++)
            list.Add(i);
        ArrayShuffler.Shuffle(list);
        return list.ToArray();
    }
    private void DisplayScore()
    {
        questionScreen.SetActive(false);
        scoreScreen.SetActive(true);
        scoreText.SetText(score.ToString());
        scoreText.gameObject.SetActive(true);
    }
    public void ResetQuiz()
    {
        curQuestion = 0;
        scoreScreen.SetActive(false);
        scoreText.gameObject.SetActive(false);
        if (curQuestion < qs.Count)
            DisplayQuestion(qs[curQuestion]);
    }
}
public static class ArrayShuffler
{
    private static System.Random rng = new System.Random();
    public static void Shuffle<T>(this IList<T> list)
    {
        int n = list.Count;
        while (n > 1)
        {
            n--;
            int k = rng.Next(n + 1);
            T value = list[k];
            list[k] = list[n];
            list[n] = value;
        }
    }
}
```

Question Base

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
//to create categories, create a category SO which stores a list of
questions
[CreateAssetMenu(menuName = "Scriptables/Question")]
public class QuestionBase : ScriptableObject
{
    [TextArea(1, 12)]
    public string question;
    public string correctAnswer;
    [Tooltip("If longer than max, a random selection will be picked")]
    public string[] incorrectAnswers;
}
```

Question

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

[System.Serializable]
public class Question
{
    public string question;
    public string correctAnswer;
    public List<string> answers = new List<string>();
    public static Question ConvertBase(QuestionBase qBase)
    {
        Question q = new Question
        {
            question = qBase.question,
            correctAnswer = qBase.correctAnswer,
            answers = qBase.incorrectAnswers.ToList()
        };
        return q;
    }
}
```

Answer

```
using TMPro;
using UnityEngine;

public class Answer : MonoBehaviour
{
    public QuizManager manager;
    TextMeshProUGUI text;
    bool isCorrect;
    private void Awake()
    {
        text = GetComponent<TextMeshProUGUI>();
    }
    public void SetNewAnswer(string text, bool isCorrect)
    {
        this.text.SetText(text);
        this.isCorrect = isCorrect;
    }
    public void SendAnswer()
    {
        manager.AnswerQuestion(isCorrect);
    }
}
```

Game Event

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

[CreateAssetMenu]
public class GameEvent : ScriptableObject
```

```
{
    private List<GameEventListener> listeners =
        new List<GameEventListener>();

    public void Raise()
    {
        for (int i = listeners.Count - 1; i >= 0; i--)
            listeners[i].OnEventRaised();
    }

    public void RegisterListener(GameEventListener listener)
    { listeners.Add(listener); }

    public void UnregisterListener(GameEventListener listener)
    { listeners.Remove(listener); }
}
```

Game Event Listener

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

public class GameEventListener : MonoBehaviour
{
    public GameEvent Event;
    public UnityEvent Response;

    private void OnEnable()
    { Event.RegisterListener(this); }

    private void OnDisable()
    { Event.UnregisterListener(this); }

    public void OnEventRaised()
    { Response.Invoke(); }
}
```

Notification Manager

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;
public class NotificationManager : MonoBehaviour
{
    public static NotificationManager instance;
    private void Awake()
    {
        if (instance != null)
        {
            Debug.LogError($"2 instances of class Notification Manager:
{gameObject.name} and {instance.name}");
            Destroy(instance);
        }
```

```
        instance = this;
    }
    private enum AlertStatus
    {
        None,
        Queued,
        Active
    }
    public CanvasGroup group;
    public LayerMask layers;
    public AudioSource audioSource;
    public TextMeshProUGUI textDisplay;

    public float fadeDuration = 0.3f;
    [Header("ReadOnly")]
    [SerializeField]
    private float displayTime;
    [SerializeField] private AlertStatus alertStatus = AlertStatus.None;

    public Transform player;

    private List<NotiDetails> queue = new List<NotiDetails>();

    /// <summary>
    ///
    /// </summary>
    /// <param name="alert">Message</param>
    /// <param name="displayTime">Min Value is 0.5f, Max is 10f</param>
    public void DisplayNotification(string alert, float displayTime = 4.3f)
    {
        if (alertStatus != AlertStatus.None)
            queue.Add(new NotiDetails(alert, displayTime));
        else
        {
            audioSource.Play();
            textDisplay.SetText(alert);
            this.displayTime = Mathf.Clamp(displayTime, 0.5f, 10f);
            alertStatus = AlertStatus.Queued;
        }
    }
    bool fading = false;
    void Update()
    {
        if (alertStatus == AlertStatus.Queued)
        {
            Ray r = new Ray(player.position, player.forward);
            ;
            if (Physics.Raycast(r, out RaycastHit hit, 5.0f, layers))
            {
                if (hit.transform != null && hit.transform.name ==
"TabletLookAtCollider" && !fading)
                {
                    alertStatus = AlertStatus.Active;
                    StartCoroutine(FadeTo(1.0f, fadeDuration));
                    displayTime += Time.time;
                }
            }
        }
```

```
        else if (alertStatus == AlertStatus.Active)
        {
            if (Time.time > displayTime - fadeDuration && !fading)
            {
                StartCoroutine(FadeTo(0.0f, fadeDuration));
            }
            if (Time.time > displayTime)
            {
                alertStatus = AlertStatus.None;
            }
        }
        if (alertStatus == AlertStatus.None && queue.Count > 0)
        {
            DisplayNotification(queue[0].alert, queue[0].displayTime);
            queue.RemoveAt(0);
        }
    }
    [ContextMenu("SendTestNoti")]
    public void SendTestNoti()
    {
        DisplayNotification("Test notification!!!", 4.3f);
    }
    IEnumerator FadeTo(float target, float duration)
    {
        fading = true;
        float startV = group.alpha;
        for (float t = 0.0f; t < duration; t += Time.deltaTime)
        {
            group.alpha = Mathf.Lerp(startV, target, t);
            yield return null;
        }
        group.alpha = target;
        fading = false;
    }
}
```

Noti Details

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class NotiDetails
{
    public string alert;
    public float displayTime;
    public NotiDetails(string alert, float displayTime)
    {
        this.alert = alert;
        this.displayTime = displayTime;
    }
}
```

Int Event

```
using System.Collections;
using System.Collections.Generic;
```

```
using UnityEngine;
using UnityEngine.Events;

[System.Serializable]
public class IntEvent : UnityEvent<int> { }
```

Hands Free Locomotion

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

[RequireComponent(typeof(CharacterController))]
public class HandsFreeLocomotion : MonoBehaviour
{
    public LocoArrowController arrowController;
    protected CharacterController cc;
    public Vector3 center = Vector3.zero;
    public Vector3 boundarySize = Vector3.zero;
    public float axisZoneDistance = 1.0f;
    public float safeZoneRadius = 0.2f;
    public float speedModifier = 10.0f;
    public float currentSpeed = 0.0f;
    public Vector3 moveAxis;
    public float lastHeadHeight;
    public float maxHeadHeight = 1.0f;
    bool moveCenter = false;

    protected OVRCameraRig CameraRig = null;
    bool playerControllerEnabled;
    void Start()
    {
        OVRCameraRig[] CameraRigs =
gameObject.GetComponentsInChildren<OVRCameraRig>();

        if (CameraRigs.Length == 0)
            Debug.LogWarning("OVRPlayerController: No OVRCameraRig
attached.");
        else if (CameraRigs.Length > 1)
            Debug.LogWarning("OVRPlayerController: More then 1 OVRCameraRig
attached.");
        else
            CameraRig = CameraRigs[0];

        cc = GetComponent<CharacterController>();
    }
    private void OnEnable()
    {
        OVRManager.HMDMounted += SetBoundary;
    }
    private void OnDisable()
    {
        OVRManager.HMDMounted -= SetBoundary;

        if (playerControllerEnabled)
```

```
        {
            if (CameraRig != null)
            {
                CameraRig.UpdatedAnchors -= UpdateTransform;
            }
            playerControllerEnabled = false;
        }
    }
    void SetBoundary()
    {
        if (OVRManager.boundary.GetConfigured())
            boundarySize =
OVRManager.boundary.GetDimensions(OVRBoundary.BoundaryType.PlayArea);
        else
            boundarySize = Vector3.one;
        axisZoneDistance = (Mathf.Min(boundarySize.x, boundarySize.z) /
2.0f);
        moveCenter = axisZoneDistance != Mathf.Clamp01(axisZoneDistance);
        if (axisZoneDistance < 0.5f)
            safeZoneRadius = 0.0f;
        axisZoneDistance = Mathf.Clamp(axisZoneDistance - safeZoneRadius,
0.0f, 1.0f);//IDK WHY /2.0F AGAIN, MATHS IS HARD
    }
    void Update()
    {
        //axisUIController.SetInfo((1.0f / Time.deltaTime).ToString());
        if (!playerControllerEnabled && OVRManager.isHmdPresent)
        {
            if (OVRManager.OVRManagerinitialized)
            {
                if (CameraRig != null)
                {
                    SetBoundary();
                    lastHeadHeight =
CameraRig.centerEyeAnchor.localPosition.y;
                    CameraRig.UpdatedAnchors += UpdateTransform;
                }
                playerControllerEnabled = true;
            }
            else
                return;
        }
    }
    void UpdateTransform(OVRCameraRig cameraRig)
    {
        if (OVRInput.GetDown(OVRInput.Button.Start))
            maxHeadHeight = cameraRig.centerEyeAnchor.localPosition.y;

        if (axisZoneDistance > 0.1f)//axisZoneDistance wrong?
        {
            Transform headTransform = cameraRig.centerEyeAnchor;

            if (headTransform.localPosition.y < maxHeadHeight - 0.15f)
                return;

            moveAxis.x = GetAxisValue((headTransform.localPosition.x -
center.x) * 2.0f / axisZoneDistance);
            moveAxis.z = GetAxisValue((headTransform.localPosition.z -
```

```
center.z) * 2.0f / axisZoneDistance);


//axisUIController.SetAxisValues(headTransform.InverseTransformDirection(mo
veAxis));
            arrowController.SetVel(moveAxis);
            float headVelocity = Mathf.Abs(headTransform.localPosition.y -
lastHeadHeight);

            cc.Move(headVelocity * Vector3.ClampMagnitude(moveAxis, 1) *
speedModifier);

            lastHeadHeight = headTransform.localPosition.y;


            if (headTransform.localPosition.y > maxHeadHeight)
                maxHeadHeight = headTransform.localPosition.y;

            if (moveCenter)
            {
                //(headTransform.localPosition.x / axisSensitivity) ==
moveAxis.x;
            }
        }
    }

    private float GetAxisValue(float x, float min = -1.0f, float max =
1.0f)
    {
        if (x < 0)
            return Mathf.Clamp(x + safeZoneRadius, min, 0.0f);
        else
            return Mathf.Clamp(x - safeZoneRadius, 0, max);

    }
}
```

Loco Arrow Controller

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

public class LocoArrowController : MonoBehaviour
{
    public CharacterController cc;
    public Material mat;
    public Color color;
    int matID;
    private void Start()
    {
        matID = Shader.PropertyToID("_Color");
    }
    public void SetVel(Vector3 fwd)
    {
        Vector3 p = cc.center;
        p.y = transform.localPosition.y;
```

```
        transform.localPosition = p;
        fwd.y = 0;
        color.a = fwd.magnitude;
        mat.SetColor(matID, color);
        if (fwd != Vector3.zero)
            transform.forward = fwd;
    }
}
```

Set New Active Disease

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

public class SetNewActiveDisease : MonoBehaviour
{
    public SelectedCondition selectedCondition;
    [HideInInspector] public ConditionProperties condition;
    public GameEvent setNewCondition;
    [ContextMenu("SetNewActiveDisease")]
    public void SetNewActiveCondition()
    {
        selectedCondition.condition = condition;
        setNewCondition.Raise();
    }
}
```

Check Hands Vs Controller

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

public class CheckHandsVsController : MonoBehaviour
{
    bool handsActive = false;
    [SerializeField] GameObject[] controllers;
    [SerializeField] GameObject[] hands;

    private void Awake()
    {

        handsActive = OVRInput.GetActiveController() ==
OVRInput.Controller.Hands;
        Toggle();

    }
    void Update()
    {
        if (handsActive)
        {
            if (OVRInput.GetActiveController() ==
OVRInput.Controller.Touch)
            {
```

```
                handsActive = false;
                Toggle();
            }
        }
        else
        {
            if (OVRInput.GetActiveController() ==
OVRInput.Controller.Hands)
            {
                handsActive = true;
                Toggle();
            }
        }
    }
    void Toggle()
    {
        foreach (GameObject go in hands)
            go.SetActive(handsActive);
        foreach (GameObject go in controllers)
            go.SetActive(!handsActive);

        if (handsActive)
        {

HandsManager.Instance?.SetHands(HandsManager.Instance.transform.Find(OVRPlu
gin.Hand.HandLeft.ToString()).gameObject,
HandsManager.Instance.transform.Find(OVRPlugin.Hand.HandRight.ToString()).g
ameObject);
        }
    }
}
```

Grabbable Controls Controller

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

public class GrabbableControlsController : MonoBehaviour
{
    #region Singleton
    private static GrabbableControlsController instance;
    public static GrabbableControlsController Instance
    {
        get
        {

            return instance;
        }
    }
    private void Awake()
    {
        // if the singleton hasn't been initialized yet
        if (instance != null && instance != this)
        {
            Destroy(this.gameObject);
        }
```

```
        instance = this;
    }
    #endregion
    private PartGrabbable grabbedObject;
    private BodyPartGrabbable bodyGrabbedObject;
    public Transform playerCamera;
    [SerializeField] int grabbingLayer;
    [SerializeField] int releasedLayer;
    Vector3 targetPosition;
    bool isGrabbing = false;
    bool inPosition = true;

    [SerializeField] GameObject[] duplicateButton;
    [SerializeField] GameObject[] detachButton;
    public void OnGrab(PartGrabbable grabbed)
    {
        grabbedObject = grabbed;
        transform.position = grabbedObject.transform.position; //maybe
calculate center of object or offset
        grabbedObject.grabbedBy.SetGrabVolumeLayer(grabbingLayer);
        isGrabbing = true;

        ToggleDuplicateButton(true);

        if (grabbed is BodyPartGrabbable grabbable)
        {
            bodyGrabbedObject = grabbable;
            ToggleDetachButton(true);
        }
        else if (grabbed.isCopy)
        {
            bodyGrabbedObject = null;
            ToggleDetachButton(true);
        }
        else
        {
            bodyGrabbedObject = null;
            ToggleDetachButton(false);//check if copy so can desroy;
        }

    }
    public void OnRelease(PartGrabbable released)
    {
        if (grabbedObject == released)
        {    //grabbedObject = null;
            if (grabbedObject.grabbedBy == null)
            { Debug.LogError("GrabbedBy = null"); return; }
            grabbedObject.grabbedBy.SetGrabVolumeLayer(releasedLayer);
            isGrabbing = false;
        }
    }

    void FixedUpdate()
    {
        if (grabbedObject == null)
            return;
```

```
        if (isGrabbing)
        {
            //Get Target position
            Vector3 size = grabbedObject.grabPoints[0].bounds.size * 0.5f +
new Vector3(0.15f, 0.15f, 0.15f);
            size.y = 0;
            Vector3 offset = Vector3.Scale(size, playerCamera.right);
//probably wrong
            targetPosition = grabbedObject.transform.position + offset;
        }
        transform.position = Vector3.Lerp(transform.position,
targetPosition, Time.fixedDeltaTime * 2.0f);
        inPosition = Vector3.Distance(transform.position, targetPosition) <
0.1f;
        transform.LookAt(playerCamera.position);
        transform.RotateAround(transform.position, transform.right, 90);
    }
    public void ToggleDuplicateButton(bool v)
    {
        foreach (GameObject go in duplicateButton)
            go.SetActive(v);
    }
    public void ToggleDetachButton(bool v)
    {
        foreach (GameObject go in detachButton)
            go.SetActive(v);
    }
    public void DuplicateGrabbedObject()
    {
        if (!inPosition)
        { Debug.Log("Not in position"); return; }
        if (grabbedObject && (bodyGrabbedObject == null ||
!bodyGrabbedObject.connected))
            grabbedObject.DuplicateHeldObject();
    }

    //handles attach, detach and destroy
    public void DetachGrabbedObject()
    {
        if (!inPosition)
        { Debug.Log("Not in position"); return; }
        if (grabbedObject.isCopy)
            grabbedObject.DestroyCopy();
        else if (bodyGrabbedObject)
            bodyGrabbedObject.ToggleDetachEvent();
    }
}
```

Blend 2 Textures Shader

```
Shader "Custom/Blend2TexturesLit"
{
   Properties
   {
      _MainTex ("Albedo (RGB)", 2D) = "white" {}
      _SecondaryTex("2nd Texture", 2D) = "white" {}
      _LerpValue("Transition Float", Range(0,1)) = 0
```

```
        _Glossiness ("Smoothness", Range(0,1)) = 0.5
        _Metallic ("Metallic", Range(0,1)) = 0.0
    }
    SubShader
    {
        Tags { "RenderType"="Opaque" }
        LOD 200

        CGPROGRAM
        #pragma surface surf Standard fullforwardshadows

        #pragma target 3.0

        sampler2D _MainTex;
        sampler2D _SecondaryTex;
                    float _LerpValue;

        struct Input
        {
                            float2 uv_MainTex;
                            float2 uv_SecondaryTex;
        };

        half _Glossiness;
        half _Metallic;

        UNITY_INSTANCING_BUFFER_START(Props)
            // per-instance properties possibly use textures here if performance while using
duplicates is bad
        UNITY_INSTANCING_BUFFER_END(Props)

        void surf (Input IN, inout SurfaceOutputStandard o)
        {
            fixed4 c = tex2D (_MainTex, IN.uv_MainTex) * _Color;
                            fixed4 c = lerp(tex2D(_MainTex, IN.uv_MainTex),
tex2D(_SecondaryTex, IN.uv_SecondaryTex), _LerpValue);
            o.Albedo = c.rgb;

            o.Metallic = _Metallic;
            o.Smoothness = _Glossiness;
            o.Alpha = c.a;
        }
        ENDCG
    }
    FallBack "Diffuse"
}
```