COMP 1682 2017-18 FINAL YEAR PROJECT

Increasing Engagement of Guided Meditation Applications with the Use of Virtual Reality

BSc GAMES DESIGN & DEVEVELOPMENT

Will Oliveira 00731609

Supervisor: Stella Doukianou

Keywords: Meditation, Applications, Apps, Engagement, Virtual Reality

Abstract

This report will aim to find out if guided meditation applications will see increased engagement with the added feature of low tech mobile based virtual reality. This report will also aim to demonstrate the reasons for this project and justify the decision made throughout all stages of development, from planning to a working prototype in the form of an Android app.

Acknowledgements

I would like to show my gratitude to my supervisor Stella for the quick responses to all my emails. Also a major thank you to Darrenlloyd Gent for his unwavering support throughout my years at university.

Also big thanks for my mum and dad for dealing with my short tempter throughout the writing and development of this project.

And thanks to Poppy for the wine.

Contents

Table and Figures

1 Introduction

1.1 Background Information

Virtual reality has evolved dramatically since its conception following rapid changes in today's technology. VR systems have seen increasing popularity over the last few years (Statista, 2018) due to the falling prices of immersive VR headsets, mobile based VR systems and advancements in miniaturization of technology (Umek and Kos, 2016) . VR has latched on to the popularity of smartphones by using their gyroscopic technology to give users a cheap and effective VR experience right at home using low tech devices like Google Cardboard.

There are many apps currently on the market that offer users a guided meditation experience to help people with stress and relaxation. However meditation and mindfulness are both skills that take regular use in order for it to be truly effective (Kabat-Zinn, 2006).

Studies show that only a fraction of all guided meditation applications available to download online today had high engagement, functionality or information quality (Mani *et al.*, 2015). Therefore this project is importance to assess the added benefits of VR and if it can help increase engagement and use. This was chosen to be a mobile application as opposed to a computer or console application so it can be directly compared with current mediation apps on the market.

1.2 Scope of the project

The scope of this project is to see if VR increases the use and engagement of a guided meditation application. A prototype Android application will be built which features meditation breathing techniques in a VR environment that the user is able to look around in. The finished prototype will then be tested by a number of users who will then fill out a questionnaire to find out if they added benefit of VR increases the apps engagement.

1.3 Project Aim

The aim of this project is to build a VR application to discover a direct correlation between VR and the engaging factor of a meditation app.

1.4 Project objectives

The main objective of the project is to have:

- Fully functional prototype
- Easy to use UI
- Three varying VR environments designed based on research
- Meditation features to help users focus on their breathing
- Effective use of 3D sound
- How to guide

1.5 Approach

Research showed that the most effective methodology for this project will be the Rapid Application Development (RAD) (P Beynon-Davies, 1999)which is an incremental model with different stages that do not need to follow a strict order. The functions of the project will be created in paralleled like mini projects and will then be combined and tested to make sure it's working each step of the way. The prototype app will be created in Unity as soon as the relevant researches into the meditation apps have been concluded. The environments will be modelled, based on research, on 3DS Max and imported into Unity Game engine where the implementation of the VR aspect will begin.

Once the prototype is complete, a variety of different users will try out the project and will be asked at the end to fill in a questionnaire to try to find out if the use of VR has increased its engagement.

2 Literature review

The aim of this literature review is to assess recently published literature by others on topics relating to VR and its current uses and what can be learnt when producing the prototype. The areas which have been researched and will be discussed are studies relating to medical and psychological therapy, VR in education and VR immersion and presence.

2.1 Research focus

The research focuses on current uses of VR in the medical and psychological field and to assess if they have shown to be effective or not and if so why. Research will then look at the past and current uses of VR in education and if it has helped with engagement or motivation. Finally the research will look at immersion and presence in VR systems. This will form the basis for the prototype to see how current VR systems are made and what consideration to take into account when designing one.

2.2 How the research will contribute to the general area

The research will aim to contribute to the general area by studying the benefits and drawbacks of VR systems and help with designing the prototype.

It aims to find out if the added feature of VR will increase the engagement and effectiveness of an app and hopefully open the door to other applications that can harness the power of VR.

2.3 Medical and psychological VR uses

Virtual reality can be said to replace sensory information from the physical world. Users are able to replace their sense of presence in the world for one in a virtual world. This makes it possible to create new environments for rehabilitation designed to improve a patients focus and compliance with the therapy (Romano, 2005).

Firstly, studies that have seen positive use of VR in pain reduction were examined. A study by doctors at the Wuhan Medical Care Centre for Women and Children in Hubei, China experimented with a VR distraction system for children to use during burns dressing changes on lower limbs (Hua *et al.*, 2015). The research concluded that the distraction from the painful procedure helped alleviate the patient's pain. A similar study that was concluded also had the same results (Chan *et al.*, 2007). These studies both argue that the difference in the child's response to pain is based on the nature of presence and distraction that the VR was producing. This shows that VR can go far beyond just entertainment purposes as to distract patients from physical pain based on their immersion in the virtual world and distraction from the real world pain.

Secondly a journal article which focused on Two Virtual Reality Pilot Studies for the Treatment of Paediatric CRPS (Won *et al.*, 2015) showed VR immersive techniques increased relaxation and engagement during paediatric patients. It argues that VR therapy is safe and feasible as patients were too concentrated and immersed in the virtual world to feel as much pain as the patients did in standard physical therapy. It also suggests that a patients feeling of presence in the virtual world not only helped distract patients from the pain but also was found to produce relaxation and increased engagement in the physical therapy. This argues that a VR experience with high presence is needed to increase engagement and to produce relaxation.

Studies also show that the entertaining nature in which VR systems are designed are seen to be more interesting and enjoyable then regular therapy tasks (Laver *et al.*, 2012). This, again, can be said to distract patients from the physical pain as they are immersed in a virtual world and concentrating on the tasks in the environment.

Furthermore, other studies support the idea that VR can be used for not only physical pain but also psychological trauma. Research by professors in the University of Valencia showed that virtual environments containing various slow breathing and mindfulness exercises were effective at increasing joy and relaxation in elderly users (Baños *et al.*, 2012). Based on existing evidence that supports the value of addressing positive emotions in order to help enhance people's mental and physical health (Tugade, Fredrickson and Feldman Barrett, 2004) it can be said that people who experience positive emotions will have an increased sense of calmness and optimism, along with higher quality personal relationships. This adds into the overall benefits of the prototype to give users an immersive mindfulness experience with the added VR element to increase its engagement.

VR therapy has shown positive results when used to treat various anxiety disorders such as phobias and the fear of flying. Studies on people suffering from post-traumatic stress disorders being treated with VR exposure based therapy were looked at. The results showed that if VR systems followed clinical guidelines for evidence-based interventions in the treatment of PTSD they would show increased results on patients (Botella *et al.*, 2015). This is due in part to an increased control over feared or anxiety causing stimuli within the virtual environment (Rothbaum *et al.*, 1995). One of the major benefits of a virtual world is that it can be anything it needs to be to help users deal with their issues.

Research into treatment of PTSD presented in a systematic review of the efficacy of virtual reality exposure therapy (Gonçalves *et al.*, 2012) suggested the potential effectiveness of virtual reality exposure therapy. It suggests it is as reliable as standard exposure therapy. The research paper also suggests VR exposure therapy can be useful in the treatment of PTSD that isn't responding to traditional exposure therapy as it allows a greater engagement by patients if a high enough level of presence is achieved.

2.4 Educational VR

VR has also emerged as a useful tool for the study and the assessment of cognitive processes and functional abilities (Buckwalter and Rizzo, 1997; Parsons *et al.*, 2007; Bailenson *et al.*, 2008). VR has a capacity to create unique and dynamic environments for learning which allows it to be assessed and studied in ways that traditional learning methods are unable to do so.

VR training simulations with head mounted displays were developed on Wright-Patterson Air Force Base in Ohio around the 60s and 70s (Spector *et al.*, 2014) and have evolved into a range of VR solutions for educational needs. Current systems are integrating into modern teaching methods due to an increase in availability and reduction of development costs and have shown the ability to increase

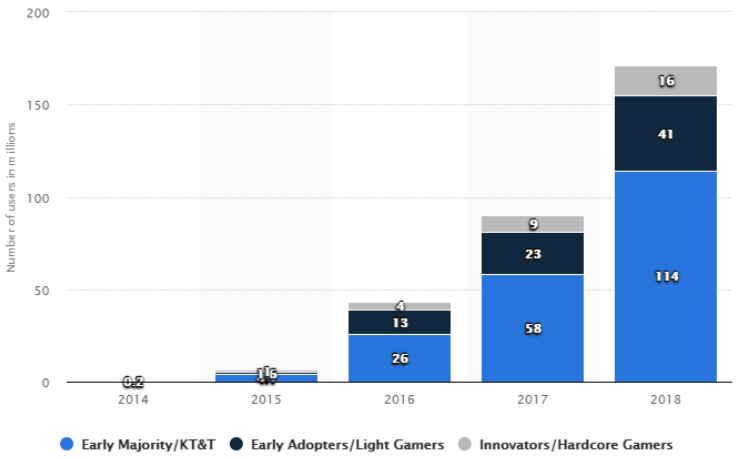


Figure 1: Statistic depicting number of active VR users worldwide (in millions)

academic performance and engagement by students (Beijing Bluefocus E-Commerce Co., 2016).

Firstly, studies have shown that a blended learning environment comprised of standard educational learning, web and mobile based learning can provide a highly effective way to educate students (Gay, 2010; Yen and Lee, 2011). Studies have shown that a VR educational tools effectiveness is based on the participants familiarity to the user interface, and how easy it was to navigate and use (Virvou and Katsionis, 2008). These results show that a systems usability and likeability is related to its familiarity and as VR use has increased exponentially in the last four years (Statista, 2018a) this could provide a standard method of education in the future. The prototype will be focused on being as easy to use and intuitive as possible by having the menu following simple standardised UI methods. This will make it easier to pick up and use by even the most novice of VR users. The main focus of the project will be engaging people to participate in meditation techniques with the added benefit of an immersive VR experience, and to compare the added benefits to VR against its drawbacks.

Secondly studies also show the increased benefits in engaging students using virtual reality-based instructions. (Merchant *et al.*, 2014) demonstrated that simulations and virtual worlds were effective in improving learning outcome gains in teaching students in higher education. This study however does state that development of desktop based virtual reality involves financial costs and trained teachers for the system to be used effectively.

Thirdly, studies have also seen increased motivation in using VR systems by children (Harris and Reid, 2005). Using a Paediatric Volitional Questionnaire (PVQ) to measure children's motivation participants used a VR play experience. Results found that children with cerebral palsy indicated high motivation while using VR and argues that VR has potential to being a successful intervention tool.

2.5 Immersion and presence in VR

A successful immersive experience needs to be graphically proficient to trick the user's brain into believing they are actually in the environment. A poorly designed and rendered environment might not have the ability to make the users brains believe they are actually there. Immersion and presence are defined as the objective level of sensory fidelity of a VR system and a user's subjective psychological reaction to using a VR system (Mel Slater, 2003). The level of a VR systems immersion is dependent on its rendering software and display technology. One VR system can have different levels of immersion than others as its immersion is objective and can be measured.

Presence however can be subjective to the users and their own personal experience and is related to the feeing of the user "being there". The level of presence in the same VR system can differ from person to person depending on their familiarity of VR and also external factors like their state of mind or recent history (Bowman and McMahan, 2007). The project will aim to create a realistic setting for the user while balanced with the graphical capability of a smartphone based virtual experience. This project is designed to be used by users every day and any time they choose so a graphic heavy photorealistic environment will be beyond the processing capabilities of most smartphones.

2.6 Conclusion

This section covered published literatures relating to VR systems and how they have been used to help patients deal with different types of physical and psychological ailments. Their research shows the effectiveness of these systems and is the basis of this project.

The use of VR has evolved from standard entertainment and training systems to effective therapy and rehabilitation tools. All studies agreed with the benefits of VR and how to best implement it to achieve the best results. The prototype is hoping to achieve increased engagement in meditation, by creating a virtual world with researched based calming and mindfulness methods. The research suggests that a virtual environment helps engage the users more than a non VR experience.

Studies also agree that a certain level of presence is necessary in order to achieve a positive outcome so the prototype has to take into consideration the environments quality and immersion.

3 Existing product review

Guided mediation apps are available on both Google play and the App Store. VR mediation apps are also available to download from both stores. This section will review the most popular apps on the market in their respective category found solely on the App Store. The purpose of this section is to understand how the standards that these apps have adopted and what lessons can be learnt in the design of the prototype.

3.1 Mindfulness and meditation applications

3.1.1 Calm

Calm (Acton Smith and Tew, 2012) is the most popular mindfulness and wellness based app currently on the market. With 200,000 5-star reviews it has been named Apple's App of the Year and Editors' Choice by Google Play. It offers a guided mediation experience with natural visuals and sounds. Calm has a range of different features from calming music, guided meditation, masterclasses, and sleep which plays out stories to help users fall asleep. It features a simple consistent UI with a navigational bar anchored to the bottom of the screen. The music page offers calming music for the user to listen to with a range of different categories. The meditate page lets the user chose a type of meditation to be guided through while playing music and showing a nature video. Masterclass has a list of videos to help users deal with issues such as breaking habits and social media addition. The sleep page shows different bedtime stories to listen to and who they were narrated by. All sounds and videos are able to be downloaded to be used offline. To unlock calm and receive the full benefits of the app including over a hundred different guided meditations and more bed time stories to listen to, users have to pay £35.99 a year after a 7 day free trial.

3.1.2 Headspace

Headspace (Puddicombe and Pierson, 2010) offer guided meditation session to help users deal with a range of issues. Headspace recommends sessions to the user based on the time of day and what sessions the user has already completed. Packs can also be unlocked, if you are a subscriber to Headspace, which contains a number of sessions. Headspace charges £9.99 a month, £74.99 per year or a onetime lifetime payment of £399.99. The sessions are narrated audio files that help users to focus their mind. Users are also able to track their progress and check their stats on their profile page. The navigation bar at the top lets users go onto kid sessions designed for kids aged 5 and under, and the animation page shows users animations along with guided self-help audio.

3.1.3 Pacifica

Pacifica (Beermann and Goettel, 2014) is a self-help app for managing stress, anxiety and depression. According to their website Pacifica is based on cognitive behavioural therapy combined with different relaxation and wellness techniques (*Pacifica - About Us*, 2018). It offers an easy to use UI

that asks how you are feeling today and lists tools that the user can use. These tools include meditate, health, hope, thought, goals and mood, all of which ask different questions to the user about how much sleep they have been getting or what has been on their mind. The guided path page gives different guided meditation exercises to help users. All activities that the user has done can be seen on their history page and a community page also allows user to find peer support from other Pacifica members. To access all guided exercises you need to sign up to Pacifica Premium which has a monthly cost of £7.99, yearly cost of £47.99, or a lifetime price of £199.99.

3.2 Mindfulness and meditation application review

This section will review the above-mentioned applications based on their UI, features and relevance to a guided meditation application.

3.2.1 Calm

It's easy to see why Calm was voted App of the year and Editors Choice by both Apple and Google as its strengths are its high usability and range of features. The UI is clear and easy to use and its and it still has plenty of features to use for free without paying for a subscription. The app also offers customisation of the home screen to different scenes which is a good feature. One of the drawbacks of Calm is the user profile which doesn't offer as much recorded data as other apps do making it hard to see what guided programs you have already used.

3.2.2 Headspace

Based on research Headspace has the highest median score on the Mobile Application Rating Scale (Mani *et al.*, 2015) which is an accurate measure of an applications visual aesthetics, engagement, functionality and information quality. This is due to its easy to use UI and clear theme. This aids to the effectiveness of its features of guided meditation as the engagement links in with the apps usability and frequency of use. A weakness with the app lies in the number of features which are free to use as these are limited.

3.2.3 Pacifica

Pacifica's UI is better than all other UI reviewed. It's aesthetically pleasing and has high usability. Pacifica also offers clear graph data of user use and input which a user is able to export in a PDF format that is emailed to you. The community feature work well d can be great help for those who need to get help from others. One of the downsides of this is the community feature can be open to misuse, however you are clearly able to report an inappropriate post by flagging it.

3.3 VR meditation applications

There are several apps on today's market that offer users a VR based mindfulness experience. The following apps were the top three app found on the Apple App Store when searching for Virtual Reality Meditation.

3.3.1 Provata VR

Provata VR ('Provata Health', 2016) is the highest rated app on the App Store when searching for VR meditation at an average rating of 4.5 based on 29 reviews. It offers users a guided meditation experience with several scenes that a user can experience and different targeted guided meditation exercises designed for anxiety, focus, calm and others. The scenes are real world videos in which a user can look around while being guided by a voice on how to breathe and tells the user different relaxation exercises. There are only four scenes which are free to use (Coral Reef, Garden Falls, Oceans Edge, and Costal Ponds) with the rest being unlocked for a payment of £3.49 per month or £32.99 per year. It offers an aesthetically pleasing and easy to use app with different settings for the

user to choose from like meditation type and length from two to ten minutes. The user can chose whether they want a full screen scene or a split screen experience which is suitable for VR cardboard devices. It has not been updated since March 30th 2017.

3.3.2 Mindspace

Mindspace (Matthias Koller, 2016) is a VR mindfulness app that provides a VR environment for the user to look around in with the goal of elevating self-awareness. Mindspace does not offer any guided meditation features and is lacking any menu system or meaningful UI. This app only provides one scene to look around in and is free on the app store.

3.3.3 VR Meditation, Yoga, ASMR

VR Meditation, Yoga, ASMR provides users with a VR videos to watch. Users are able to watch a range of different VR videos from yoga, sports, horror, and entertainment. The UI is VR based so the user has look in the environment for the scene or options they want to view or press then select it via a VR cardboards input button.

3.4 VR Meditation application review

This section will now review the aforementioned VR meditation applications based on their UI, features and relevance to a VR meditation application.

3.4.1 Provata VR

Provata has the best looking app of all VR meditation apps reviewed and is the closest thing to a VR meditation experience that was found on the App store. It has a number of free to use scenes and a range of features to help users deal with different issues. However, Provata VR's are just videos that a user can look around in using a VR headset. This loses the advantages that a custom designed 3D scene has which can be made to certain specifications. The videos also affect the apps file size with just the basic app with four scenes being 538.6mb. This can make it difficult for people to download it as they might not have enough memory on their phones for the app or new scenes. It also lacks any meaningful reviews to judge if the use of VR has made people more likely to use it regularly.

3.4.2 Mindspace

Mindspace shows a scene that works well with a VR cardboard headset and looks good. However, the app lacks any real menu system with just a simple UI which the user has to look down to use and they switch the VR mode on and off, re-centre the scene and reset the scene. This app offers not guided meditation and just lets users see a scene with background music. It also lacks a transitional scene to let users place their phones into a VR Cardboard viewer

3.4.3 VR Meditation, Yoga, ASMR

VR Meditation, Yoga, ASMR has a range of videos to watch some of which show guided yoga and meditation exercises. This makes this app less of a guided VR meditation application and more of a VR video library. The UI can only be used while in the VR mode making it complicated and challenging and there is no transitional screen to let users put their devices into a VR Cardboard viewer.

3.5 Conclusions

The existing meditation apps have shown to be an effective way for people to practise mindfulness and meditation at home. Reviews have described the positive effects these apps have had on people's lives and that they are more than just a novelty. The numbers of features these apps have also make it easy for a user to come back and try out different exercises and techniques.

The VR apps follow a trend of unpopularity towards mobile VR applications with it holding the smallest percentage of VR applications (Statista, 2018). This means they are outdated and not completely fit for purpose. They do show that meditation techniques can be incorporated into VR whether that is in a 3D environment or a VR video. There is however a lack of real meditation applications based in a virtual world and the ones that are very basic and nowhere near as extensive as the features that are offered by their non-VR alternatives.

4 Technical Review

This section will review the software that will be used when creating their prototype. This includes a list of game engines that were considered for the development of the prototype and the compiler that will be used to code the app.

4.1 Game Engine

To create the prototype, a game engine was needed to design all aspects of the application. This list compiles the most common game engines that are available on the market. A review of their features and familiarity was made to find the best suitable match to create the prototype (see Table 1).

Name	Price	Difficulty	Mobile Platforms	Familiarity
Unreal	Free	Professional	Android, iOS	Used Rarely
Unity	Free, £26/month Plus, £93/month Pro	Intermediate	Android, iOS, Windows Phone, Tizen, Fire OS	Used Extensively
CryEngine	£7.43/month	Professional	None	Never Used
GameMaker: Studio	Free, £28, £71, £286	Intermediate	Android, iOS	Never Used

Table 1: Game Engine Review

As the prototype will be mobile based as a platform based application wouldn't be in keeping with the engagement aspect of the game, CryEngine was ruled out. Unity will be used on the project due to extensive personal use of the software and its wide range of features and compatibilities with most mobile devices. Unity will be used to create the 3D environment and features of the prototype including UI.

4.2 IDE

An IDE program was required in order to write all the code of the prototype. The IDE needed to be easy to use and able to integrate with Unity. Since C# was chosen as the programming language of the app due to greater personal knowledge of it compared to Java, the IDE must be compatible with it.

Name	Unity integration	Supported languages	Ease of use
MonoDevelop	Yes	C#, F#, Visual Basic .NET, C/C++, Vala, Java, Scala, Boo, Nemerle, Python, JavaScript, Oberon, PHP, Object Pascal, Lua, Cobra, Synergy-DBL and others	Simple interface and designed to be integrated with unity
Visual Studio 2017	Yes	Batch, C#, C++, Clojure Coffee Script, CSS Dockerfile, F#, Go, Groovy,	IntelliSense and easy code navigation makes it very easy to work in

		Jade, Java, JavaScript, JSON, HandleBars, HTML Ini, Less, Lua, Makefile Markdown, Objective-C, Perl, PHP, PowerShell, Python, R Razor, Ruby, Rust, Sass, SQL, Swift, TypeScript, Visual Basic, XML and others	
Android Studio	Yes, with extension	C++, C, C#, SQL, Java, JavaScript, HTML5, Python, Perl, Ruby, PHP and others	Built by Google for Android app development, Unity integration is possible but not as easy as the other IDE's

Table 2: IDE Review

Visual Basic was chosen because of its ease of use and also because it supports C#. To integrate Visual Basic with Unity, the external script editor was switched from MonoDevelop to Visual Basic in the Unity preferences options.

5 Requirements Elicitation

This section covered the functional and non-functional requirements that were to be expected from the prototype in order for it to accurately assess the benefits of virtual reality.

5.1 Functional requirements

Functional requirements are the behaviours that are expected from a system. These convey the functionalities that are a priority for the product to work. The following requirements were chosen as they are imperative to creating a working engaging product.

- The frame rate must be high enough to give the effective illusion of movement
- User to be able to browse the different environments and breathing techniques
- The user needs to be able to look around the 3D environment with accurate head tracking
- Effective 3D sound throughout the scene to increase users immersion
- The user needs to clearly see breathing guide to keep their breathing consistent to their chosen setting
- Once finished, or when the user wants to, they should be able to exit the 3D environment
- The product must work effectively using Google Cardboard or other low tech VR devices.

5.2 Non-functional requirements

The main requirements for this project are engagement and immersion. To facilitate both those aspects, non-functional requirements are needed as they are the architectural qualities of a functioning product. These are:

- An effective UI based on research into calming colours, typography and consistency throughout
- The application should be stable enough to not crash in use as this will effect results and ruin engagement
- The user needs to be able to open the app and start a meditation session within 10 seconds
- Bugs should be completely eliminated as they will lower effectiveness of the product and its engagement

6 Legal, Social, Ethical and Professional Issues and Considerations

Copyright law was looked into to make sure all work done was completed legally. Certain features from other apps previously mentioned will be used without the theft of anyone's intellectual property. All lines of codes in the product are not be copied form any other app or any other software to avoid infringing on any copyright. UI will be designed to fit with the product but not be based on any existing application. UI colours will also be taken into consideration as to not be difficult to use or read with anyone suffering from any sort of visual impairment.

The 3D environment won't be based on any building or town and will feature natural looking settings and consideration will be taken to not have any flashing images that could affect people with epilepsy.

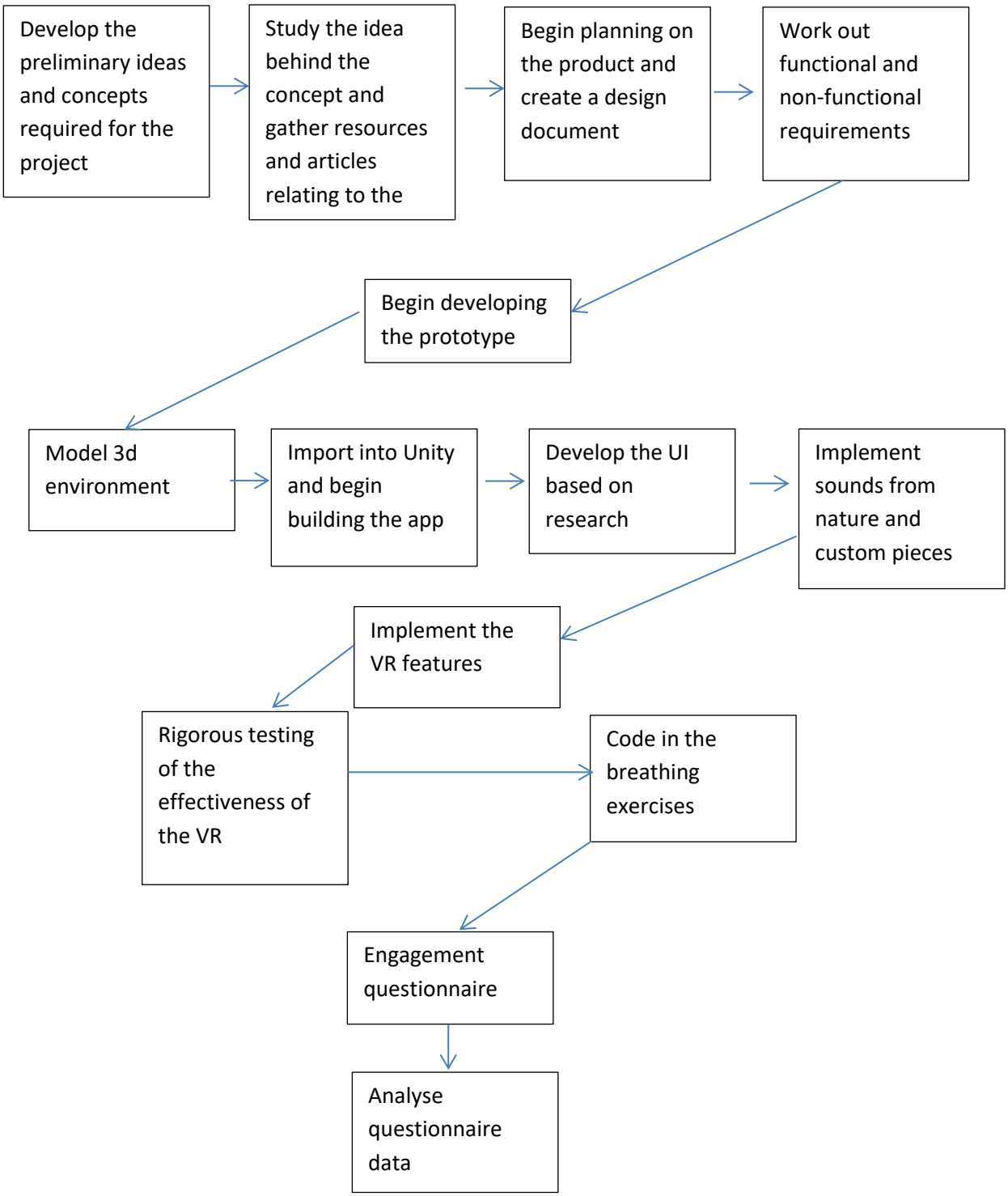
7 Design

7.1 Introduction

This section will provide a detailed design documentation of the prototype which will be created for this project. The design documentation mentions the main aspects which were explored before the prototype was started.

The design will go through how and why each scene was designed and how the user interface was made and how it works and guides the user. Focus will then be on the scripts in the prototype that were made and why they were needed. This hopes to highlight the design considerations that were taken to ensure a working prototype.

7.2 Critical path analysis



7.3 Scenes

Research dictated that for a VR system to have a successful amount of engagement the user needs to have a high sense of presence within the environment. This made it important to create scenes that immersed players and are realistic enough to produce the needed amount of presence.

Scenes were designed to have a natural calming setting where the user is able to look around and be immersed in the scene. The scene will not offer any interaction as the only thing the user should be concentrating on is breathing and mindfulness.

The scenes were originally going to be modelled using 3DS Max to be in as much detail as it could but the desired look proved challenging. After a few test scenes 3DS Max modelling was abandoned and Unity terrain tools were used instead (see Figure 2 & Figure 4)

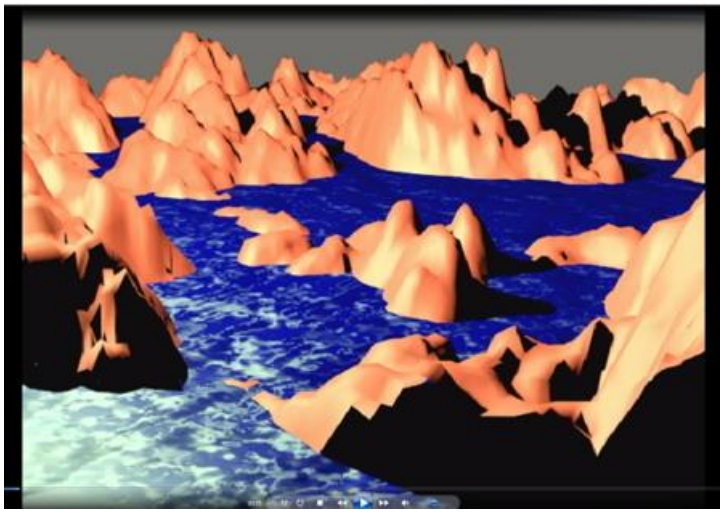


Figure 2: 3DS Max Test Scene 1

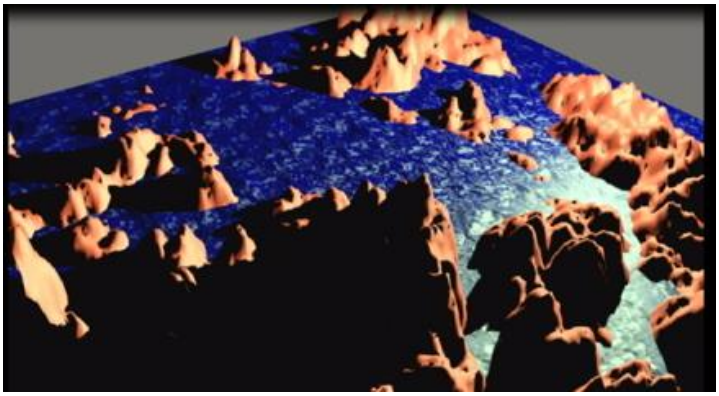


Figure 4: 3DS Max Test Scene 2

7.4 UI

The UI was designed so the user could choose a scene and quickly enter it without delay. The UI was to follow a blue theme in all buttons and the breath bar. Preliminary sketches were made and to design the scene selection menu (see Figure 3). This was designed to be the first scene the users saw when opening the app. The menu was designed to show pictures of the scenes which the user could scroll through to pick the one they wanted. They would then be asked to pick their breath length and then enter the scene.

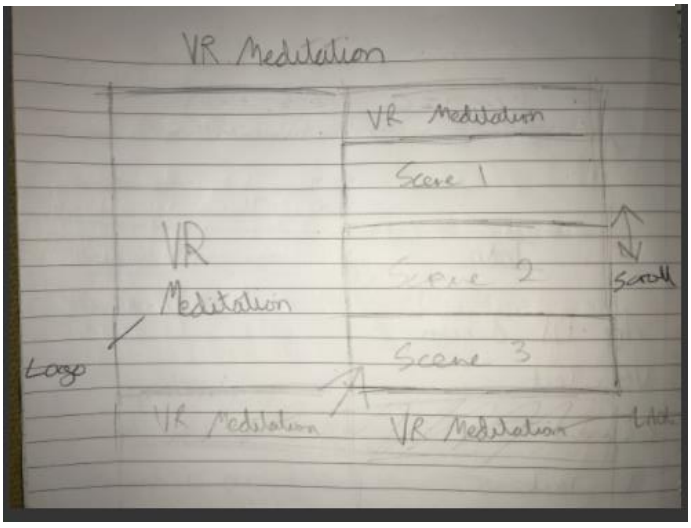


Figure 3: Menu Sketch

7.5 Font

After research on what would be the best font to use for the app the font Ubuntu was chosen. This was because it has been regularly used in apps and was an easy to read sans-serif which had been designed specifically for clarity on desktop and mobile screens (see Figure 5).

Figure 5: Ubuntu

8 Implementation

Using standards set out in Unity game development essentials (Goldstone, 2009) the first scene, Mountains, was created using the terrain tools (see Figure 6). A grass texture was then applied along with its corresponding bump map. The scene was designed in a way to have the user do a lap round the main mountain and be able to look around and enjoy the scenery. A basic water texture was added to turn the raised areas of terrain into islands. The basic water texture was used, as it being a large scene, to keep the frame rate down as much as possible. As prototype is mobile based a careful consideration was taken to ensure that it would run smoothly and give users an immersive experience which would be lost in a low frame rate environment.

Figure 6: Mountains

The second scene, Waterfall, was modelled with the terrain tools. This followed an opposite characteristics to the first scene as the mountains are on the outside surrounding a lake. The user is moved from one side of the terrain to the other while passing trees and a waterfall. This again used a basic water texture to ensure a low frame rate in such a large environment.

Figure 7: Waterfall

The third scene, Moonlight River, was designed to show the effects of a night time scene. The player is taken up the river with full view of the moon. The other side of the river has a fire which the user passes by, they then carry on to the back to the right side of the river before the

Figure 8: Moonlight River

scene ends. This scene uses an advanced water mesh to get accurate reflections of the moon.

8.1 Lighting

All scenes were made using directional lights to simulate light sources in the scene such as the sun or moon. Lights were arranged in a way to highlight shadows in each terrains bumpy texture and their tint was modified to fit the look of the scene. Lights were then baked onto the scene in order to save on processing power.

8.2 Occlusion culling

Occlusion culling was used to save on processing power. Occlusion culling works by baking all objects onto my scene and only rendering them when the camera is looking at them. This was able to save on the processing that would have been wasted on rendering items in the scene out of view of the camera. All scenes used occlusion baking on their terrain and scene object to maintain a high FPS.

8.3 Sounds

Sounds were added to each scene using the Audio Play script. Sound sources on the different GameObjects would then be playing out selected nature clip with spacial blend set to 1 to ensure a 3D sound experience. The implementation of 3D sound provides the user with localized audio cues within the environment (Davis *et al.*, 2003) tricking the brain into immersion in the scene. This increase in immersion in turn increases a user's presence within the scene (Péruch and Gaunet, 1998).

All sounds were sourced online from YouTube and are free of any copyright as they belong in the public domain as stated on the user's descriptions or websites. User interface sounds were downloaded from a public domain UI pack. All sounds were edited using Audacity audio software to give them the right length for the scene. The inhale exhale sounds were recorded on an iPhone and edited with Audacity.

Each sound was carefully chosen to have significant important in creating realism in the respective scene. Scenes have bird sounds which are positioned near terrain trees so a user is able to hear bird sounds when the camera moves past. Other scenes such as Waterfall and Moonlight River have sound sources attached to their waterfall effect and fire respectively. The use is designed to pass the sounds so they get the full effect of the 3D sound. Waterfall and Moonlight River also have audio sources on their respective cameras to add sounds that are played throughout the scene and didn't need a special blend effect. Waterfalls' waterfall noise is both on the camera and on the scene to make sure it was always heard. When the camera was far away the sound from the waterfall was not loud enough so this was chosen rather than a linear or custom rolloff in the audio source settings as a logarithmic rolloff offered the most realistic sound experience.

8.4 UI

This section covers all the UI elements in the prototype and why they were chosen. These include the menu and the How To menu along with the UI for the breath bar indicator.

8.4.1 Menu

The main menu appears on the first scene of the app. After reviewing VR applications with VR UI they were found to be difficult to use. For that reason the menu appears on the phone as normal.

The menu was originally going to be a scrollable list of pictures of the scenes for the user to select (see Figure 9). However issues arose with having the menu portrait and the game landscape which after further research was found to be a problem with the Google VR SDK which wouldn't allow it without having the scenes be available in portrait. This would have not worked on the Google Cardboard so a new menu was designed.

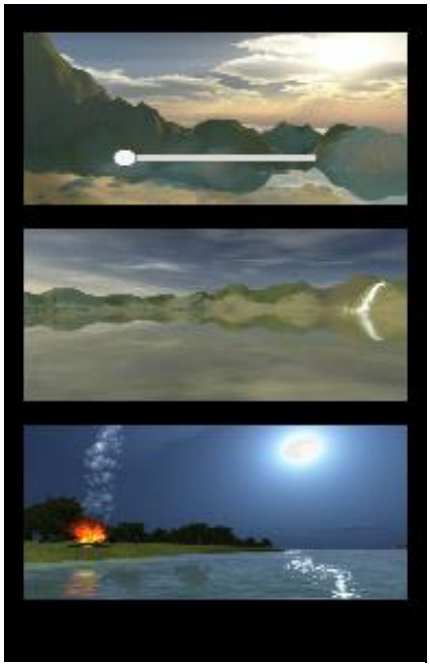


Figure 9: Original Menu

A simple background beach scene was designed to make the menu more aesthetically pleasing using a terrain with a sand texture and a Water4Advanced mesh to give it a beach look complete with waves.

The new menu used a drop down list to select which of the scenes the user would like to choose and how long they want their breath length to be. This enables the user to set a consistent breathing rate to follow which will be shown to them using the in-scene UI. The menu has a dark background with reduced opacity to allow users to see the background yet still be able to easily read the text.

The menu also has How To menu to open which guides the user on how to use the app effectively and it appears by pressing the How To button and disappears when pressing the back button. The How To was written in a text box but this didn't seem to work on the mobile tests so a JPEG file created on Photoshop with the How To text. This however gave it a loss of quality that was difficult to read. The problem was then discovered to be a wrong resolution setting on Unity and the text was once again written onto a text element with an opaque background making it even clearer to see. A function on the How To button enables it making it visible and the back button attached function disables it.

8.4.2 Font

In the original design the font Ubuntu was chosen however after mobile tests this was found to not work on some android devices as demonstrated in Figure 10. The decision was then made to use a bold Arial font to ensure compatibility on all devices and clear legibility.

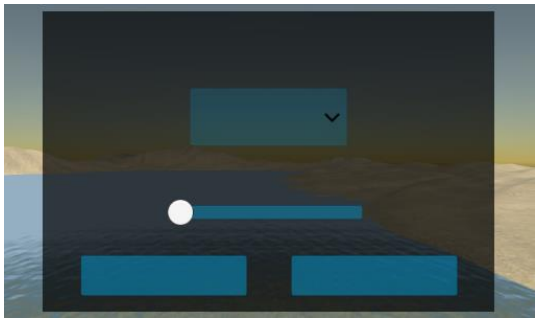


Figure 10: Font Error

8.4.3 Breathbar

The breath bar is designed to guide users on how to breathe, keeping their breathing rhythms consistent to their chosen setting. The setting is chosen from a slider on the main menu. The slider value is then translated to a Mathf.PingPong function which bounces between two floats. This bouncing float is then translated to the Breath bar's fill amount allowing it to rise and fall like a breathing rhythm. The

Breathbar is displayed to the user on their VR display with a lowered opacity to allow users to be able to still see the environment. The blue colour was chosen to match the blue UI theme set out by the buttons on the menu screen.

8.5 Virtual Reality

The virtual reality UI is provided by the Google VR SDK for Unity with Android. This is a package designed by Google for the development in unity using Google Daydream or Cardboard. The package features demo scenes which were used to see how a VR game would work on Unity. To configure the build settings and player settings to work with VR the Android platform has to be selected and the Virtual Reality support has to be enabled by going on Player Settings > Other Settings > Virtual Reality Supported. A VR device is then selected on Player Settings > Other Settings > SDKs and choosing Cardboard for the prototype.

When Cardboard is at the top of the Devices list the Unity splash screen is also in VR so the first device is set to None and the second is Cardboard. Cardboard is then loaded when the scene changes from the menu to a VR scene using the Manager script.

8.6 Skyboxes

The skyboxes texture were sources online are all in the public domain. The textures were downloaded then a skybox material was created on unity to add all skybox elements. Each one was skybox was chosen as it fit their corresponding scene the best. When created the skybox material would often have the top tile in the wrong orientation which had to be turned clockwise using Photoshop.

8.7 3DS Max

The terrain were originally going to be made on 3DS Max then imported into the scene. This proved to be more complicated than previously thought and the decision was made to stick to the Unity terrain as it is simpler to use and still gives a realistic looking scene.

3DS Max was however used to make the logs where the fire effect in the Moonlight River scene emanated from (see 8.8.2 Fire). The logs were then given a bark material to make it look more like wood.

8.8 Particle effects

To create more dynamic scene, particle effects were used on two scenes. These effects were made to look like a waterfall and a fire. They also added an extra feature to be looked at and heard with the 3D sound to further immerse the player in the scene.

8.8.1 Waterfall

The waterfall effect was achieved by a combination of four different particle effects and is featured in the Waterfall scene (see Figure 11). The first is positioned on top of a cliff edge and and uses a water splash picture. The image is then uploaded to the particle effects as a particle/additive shader. This allows the black areas of the image to be transparent. The particle is created at the top and using the gravity modifier it flows down to the water. By setting its colour over time to go transparent and its size over time to make it grow as it falls, an accurate water spray effect is created.

The second effect was another falling water effect used a cloud image with the same particle/additive shader option. This was made to look more substantial than the last effect and was to look like the bulk of the water falling from the waterfall. The settings were similar to the

WaterfallSpray effect but it didn't change its colour over time. This effect also had a smaller angle from its cone shape making it a more concentrated water flowing effect.

The third effect used the same cloud particles to look like waves coming out the water giving it more realistic effect by using a circle shape and no gravity modifier.

The last effect uses the splash particle image and was created to look like spray coming from the foot of the waterfall.

these effects were created to make the scene more visually intresting to the user. The particle effect coupled together with the 3D sound created an even greated immersive experience. the camera was made to move near the waterfall so the user could clearly see it.

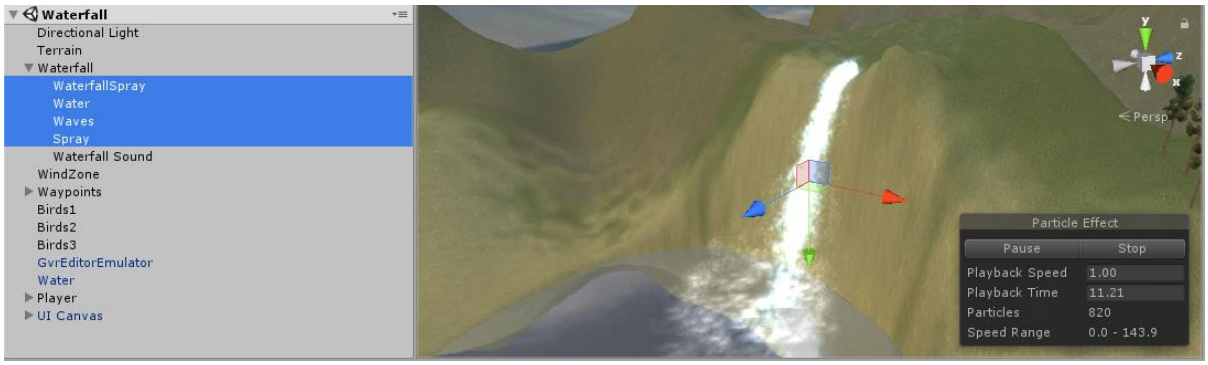


Figure 11: Particle Effects Waterfall

8.8.2 Fire

The fire effect was also achieved with four different particle effects and it is featured in the Moonlight River scene (see Figure 13). This, like the waterfall effect, was designed to add a more visually pleasing aspect to the scene.

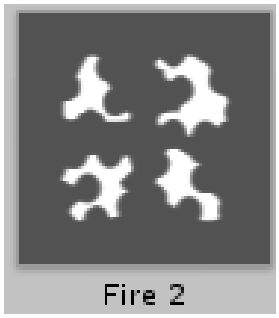


Figure 12: Fire Material

The material was first created using Photoshop with a motion blur to soften its edges (see Figure 12). They were then uploaded to Unity as a Particles/Alpha Blended shader material. The first effect takes the shapes and adds a negative gravity multiplier to make the shape go travel upwards. A colour over lifetime effect that turned it from orange to black to clear was then added to create a realistic looking fire. The same was done with the second effect using the same material. Both emanated from cone shapes of different radiuses and were given a rotation over lifetime to add a more chaotic element to the fire.

The smoke was created using the same cloud material as was used in the waterfall effect. This was then given a negative gravity multiplier and its colour was set to fade overtime. Noise was also added to add more movement to the smoke.

The last effect was made to look like sparks and embers flying out the fire. This was created using a simple particle material with a colour changing from orange to black over time to give the look of cooling down over time. Noise was also added to make them fly around the fire like real embers would.

The particle effect was then given a fire sound. This would create an immersive 3D effect when the player is moved past.

Figure 13: Particle Effects Fire

8.9 Scripts

8.9.1 Manager

The manager script is the main script in the prototype. It is set in a _Manager GameObjects with a DontDestroyOnLoad() function to prevent it being destroyed when a new scene is loaded (see Figure 14). All UI elements are set as public variables then dragged into their respective places in the unity inspector under the UI Elements header.

```
void Awake()
{
    //instructs game to try to render at 30fps
    Application.targetFrameRate = 30;

    //Check if instance already exists
    if (instance == null)
    {

        //if not, set instance to this
        instance = this;
    }

    //If instance already exists and it's not this:
    else if (instance != this)
    {
        //Then destroy this. This enforces our singleton pattern,
        //meaning there can only ever be one instance of a GameManager.
        Destroy(gameObject);
    }

    //Sets this to not be destroyed when reloading scenedestor
    DontDestroyOnLoad(gameObject);
}
```

Figure 14: Awake Method

The Awake method checks the instance of the manager script to make sure the first script is the main script and any other scripts are destroyed. This ensures that when the first Manager script is loaded it stays working throughout the app.

The Start method sets the VRSettings to false so that the menu does not have to be viewed through a VR device as a VR menu would be too complicated to use. It then adds listeners to the three UI buttons and has them do the different functions on the script.

The submit button saves the scene selected and breath length. The scene is then loaded and the breath slider value is used to determine the BreathBar frequency. A coroutine is started to load the chosen scene.

The HowToMenu and HowToMenuBack functions are designed to turn the How To menu on and off when the user selects them.

The LoadScene coroutine can be seen on Figure 15 loads the chosen scene and uses a while loop to check the progress of the scene that is loading so it can pass that information on to the load bar and show the user how much of the next scene is left to load. The cardboard device is then loaded and the VR settings are enabled. By having the cardboard be selected in the script it is possible to load the game in a non-VR mode then set it to VR when we chose a scene. Without that the Unity splash screen and menu would appear in VR when the app was opened.

```
public void HowToMenuBack ()
{
    howToUI.SetActive(false);
}

//coroutine to load the next scene and use the loading bar
IEnumerator LoadScene(int sceneIndex)
{
    //load the chosen scene and work out its progress to be set to
    //the fill amount of the loadbar
    AsyncOperation nextScene = SceneManager.LoadSceneAsync(sceneIndex);

    while (!nextScene.isDone)
    {
        float progress = Mathf.Clamp01(nextScene.progress / .9f);
        loadBar.fillAmount = nextScene.progress;
        yield return null;
    }
    //start the loadDevice coroutine to set the VR device to google cardboard
    StartCoroutine(LoadDevice("cardboard"));
    yield return null;
}
```

Figure 15: LoadScene Coroutine

The Reset function is used to reset the app to the menu when the user has finished their scene and has seen the finish scene. The Fader function uses the fading script to increase the alpha, over a period of time, of a black sprite covering the screen to give a fading effect.

8.9.2 AudioPlay & AudioSounds

The AudioPlay script is designed to make it easier and more efficient to add sound into the environment. This script allows objects and sounds to be dragged in and set at what distance to the player the sound will play along with other settings without having to add individual audio sources to each component. Is also only plays the sound if the user is at a chosen distance saving on processing audio files that are too far away or don't need to be heard yet. The AudioSounds script sets public variables that are to be used which includes a GameObject, audio clip, volume, pitch, special blend, loop setting, max distance and the trigger distance. The AudioPlay script creates a public array of the AudioSounds script called sounds which is then available to have components dragged into it via the inspector. The Awake method uses a For loop to loop through each iteration in the sounds array. It then creates an audio source for the component and adds each individual setting to its respective object.

The Start method then finds all the transform positions of each GameObject and adds them to an array. The Update method (see Figure 16) then runs a For loop for the length of the sounds array. This checks each iterations position and the player's position to compare how close they are. If their distance is under the set trigger distance then the sound is played through the GameObject if there isn't a sound already playing. If the distance is over the distance then the audio is told to stop playing.

```
// Update is called once per frame
void Update()
{
        //checks if player is under the distance so that the audio can start playing
        for (int i = 0; i < sounds.Length; i++)
        {
            playAudio = sounds[i].soundSource.GetComponent<AudioSource>();
            proximity = Vector3.Distance(transform.position, positionArray[i]);
            triggerDistance = sounds[i].triggerDistance;

            if (proximity < triggerDistance)
            {
                //checks that the audio is not already playing
                if (!playAudio.isPlaying)
                {
                    playAudio.PlayOneShot(playAudio.clip);
                }
            }
            else
            {
                //stops audio when ditance is above the chosen distance
                playAudio.Stop();
            }
        }
    }
}
```

Figure 16: AudioPlay Update Method

8.9.3 CameraMovement & Waypoints

The CameraMovement script demonstrated on Figure 17 is designed to move the player around the scene by going through different waypoints. First a number of empty GameObjects are added around the scene where the camera is going to be moving through. All Waypoints are made a child of another empty Waypoint object. Then the Waypoint script is attached to the parent Waypoint GameObject. This script iterates through every child in the Waypoints GameObject and adds their transform position to an array using a For loop.

The camera movement Start method then takes first position in the array and saves it to a target Transform variable. With the CameraMovement script attached to the camera is able to move its position towards the target. If the distance to the next target is below the chosen distance then the following waypoints transform coordinates become the target and the camera switches to move

```
// Update is called once per frame
void Update () {
    //check if the targets position compared to the cameras
    Vector3 dir = target.position - transform.position;
    //tranform the camera position to go towards the next waypoint using the speed
    //variable and Time.deltaTime
    transform.Translate(dir.normalized * speed * Time.deltaTime, Space.World);

    //check distance to next waypoint
    if (Vector3.Distance(transform.position, target.position) <= wpDistance)
    {
        //if distance to next way point is bellow the chosen distance
        //the GetNextWaypoint function starts
        GetNextWaypoint();
    }
}


void GetNextWaypoint()
{
    //checks if the next waypoint is second to last on the array
    if (waypointIndex >= Waypoints.points.Length - 2)
    {
        //if waypoint is second to last then fade out the scene and start the finishScene coroutine
        Manager.instance.Fader(1);
        StartCoroutine(FinishScene(2));
    }

    //adds a value to the waypoint index and changes the camera to follow the next waypoint
    waypointIndex++;
    target = Waypoints.points[waypointIndex];
}
```

Figure 17: CameraMovement Script

towards the next waypoint. By doing this the camera is able to smoothly move around the scene independently and the speed at which it moves and distance from the target to which it switches target can be easily modified in the inspector tab.

8.9.4 MeditationUI

The MeditationUI script is used to take the value of the user defined breath length and translate that to the Breathbar to guide users how to breathe in a slow consistent way. It works by finding the manager object by searching for its tag and then sets the breath option to equal the interval float. The script then find the image component the script is attached to and, in the Update method, uses a Mathf.PingPong function to bounce between a 0 and 1 fill amount. Using the interval divided by the Time.timeSinceLevelLoad this makes the interval length equal to the length of the PingPong going from 0 to 1 and vice versa. This keeps a consistent interval of filling the Breathbar making it a perfect guide for the user to follow by inhaling when the UI is increasing and exhaling when it is decreasing.

The second part of the Update function checks the fill amount of the Breathbar to trigger a clip from the audio source. The clips are inhale and exhale sounds which are played when the user is supposed to inhale and exhale further aiding a consistent breathing rate.

```
// Update is called once per frame
void Update () {
    //sets the timer to bounce between 0 and 1 based on the user
    //chosen interval then change the breath bars fill amount accordingly
    timer = Mathf.PingPong(Time.timeSinceLevelLoad / interval, 1);
    fillImg.fillAmount = timer;

    //when the fill amount is at the bottom play the inhale sound clip
    //when it is at the top play in exhale sound clip
    if (timer <= .1f && !GetComponent<AudioSource>().isPlaying)
    {
        gameObject.GetComponent<AudioSource>().PlayOneShot(inhale, 1f);
    } else if (timer >= .9f && !GetComponent<AudioSource>().isPlaying)
    {
        gameObject.GetComponent<AudioSource>().PlayOneShot(exhale, 1f);
    }
}
```

Figure 18: MeditationUI Update Method

8.9.5 Fading

The Fading script was made to fade between scenes giving the overall app a more professional look. Also when changing scenes sometimes the scene was loaded before the VR device was or the device would not deactivate in time. This caused VR effect in areas that were not desired such as on the menu screen for a few seconds. It works by setting a fade direction to 1 or -1 which would fade in from a black sprite covering the scene or fade out. The fadeSpeed is set by a public variable on the inspector panel and can be made faster or slower. The sprite's alpha setting is then changed by using the fade direction multiplied by the fade speed multiplied by Time.DeltaTime. Time.DeltaTime keeps the fading consistent to seconds and not to the games framerate.

8.9.6 SliderVal

The SliderVal script shows how many seconds the user has selected using the menu slider in real time. When the slider is moved its value is then sent to the SliderVal script which displays the number on its attached UI text object.

8.10 Optimization

As this app is designed for mobile use so it was imperative to make sure the game took all relevant steps to keep frame rate as high as possible and keep processing down to a minimum. To do this several setting were looked at to make sure the game would maintain a high enough frame rate to give users an accurate sense of immersion while still remaining at a high enough quality. Table 3 shows the relevant performance specifications of the computer and android device that was used for optimization testing. Optimization tips were researched and used from the Unity Optimizing Graphics Performance page on the Unity website (URL: https://docs.unity3d.com/Manual/OptimizingGraphicsPerformance.html).

The stats window on Unity shows in game performance. Frame rate the number of frames that the game is playing at per seconds. For optimal performance this should be at 60, or 30 for mobile games. Batches relate to the engine attempting to combine the rendering of objects into chuck in the computer memory. Tris and Verts relate to the number of triangles and vertices that are being rendered on the screen to make up the shapes of the scene. Finally the SetPass are the number of call that is made to the CPU to switch shader passes. The lower the SetPass calls and Batches are, the less processing power it is taking to render the scene.

Specifications	Name	CPU	RAM	GPU
Laptop	Lenovo IdeaPad U410 Touch	Intel Core i7 3537U 2.0GHz Processor	8GB	NVIDIA GeForce 710M
Phone	LG G3	Quad-core 2.5 GHz Krait 400	3GB	Adreno 330

Table 3: Laptop & Phone Specifications

8.10.1 Water

All scenes have water effects implemented into them so it was important to have the right settings for each one. The Mountain scene was used as a test scene with a still Player prefab to see what setting can be changed to increase performance. All screen shots were taken after the game had been loaded for ten seconds. All water prefabs were obtained from the Standard Assets package.

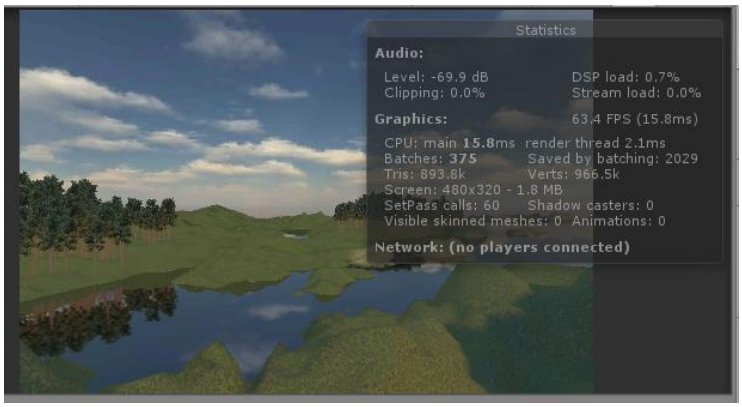


Figure 19: WaterProDaytime

8.10.1.1 WaterProDaytime

A WaterProDaytime prefab was first tested with the water mode set to refractive (see Figure 19). This gave very realistic looking water effects with good reflections and refractions. However the high quality was expensive to process with an average FPS at 60-70 and batches at 375 and SetPass calls at 60. This was far too high to work smoothly on the android test device so this high quality water was ruled out.

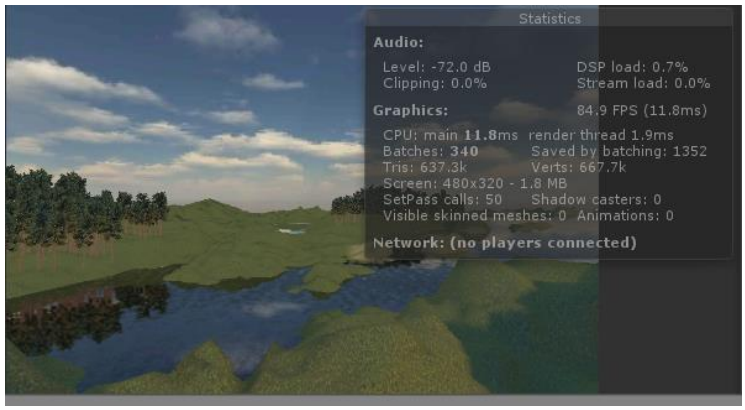


Figure 20: Water basic

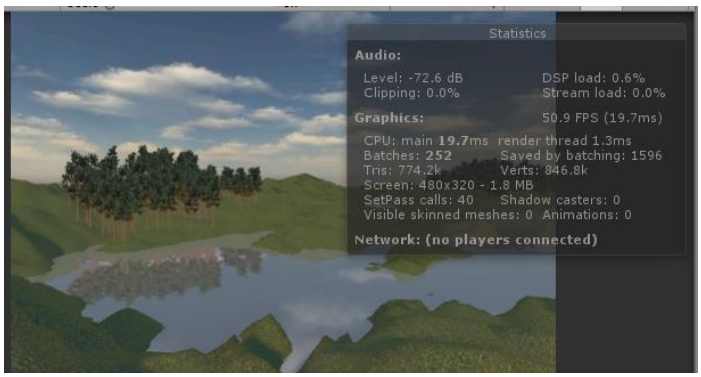
8.10.1.2 Water

Next the Water prefab was tested with a refractive water mode setting (see Figure 20). This ended up increasing the average FPS to between 80-90 on the same scene and lowering the batches to 340 and the SetPass call to 50. This gave a look just as good as the WaterProDaytime and did make a slight difference in the mobile test. However, problems arose with the VR using this water prefab as

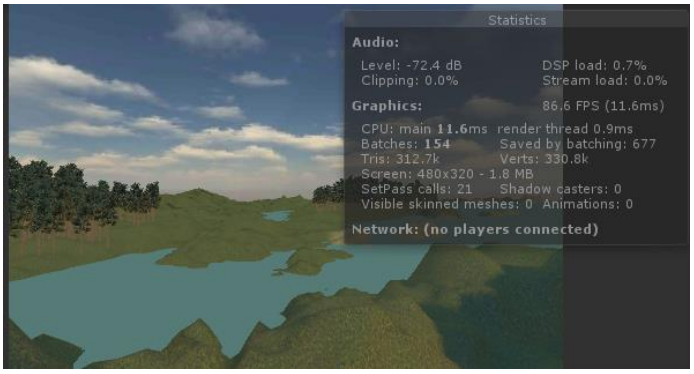
the reflections did not seem to match one one of the rendered screens (see Issues, Figure 35: Reflection Error). This made it lose the illision of Virtual Reality and lost its immersive factor. Upon further research it was discovered that this is a problem with the Water script which didn't seem to work well with VR.

8.10.1.3 WaterVR2

To overcome the reflection issue a new script was found online (Author: anubisza URL: https://github.com/googlevr/gvr-unity-sdk/issues/212) and attached to the water prefab .



The Water prefab with the new script, WaterVR2, was then tested. This script worked well on the computer texts and on the mobile test. The water reflection bug was fixed on the VR mode and it offered the best performance saving than all other tested waters. The FPS was a little lower than the other tests at around 48-55 but the batches and SetPass calls are significantly lower at 252 and 40 respectivly.



To further reduce processing the water mode was changed from reflective to simple and the stats were examined. This had a dramatic imapact with reductions in all aspects of processing. FPS went up to 80-90 and tris and verts were reduced by over half. Batches and SetPass Calls were also reduced. This setting however made the water very unrealistic looking and even though it saved on processing, too much visual quality was lost for it to be implemented into the game and still have an immersive effect.

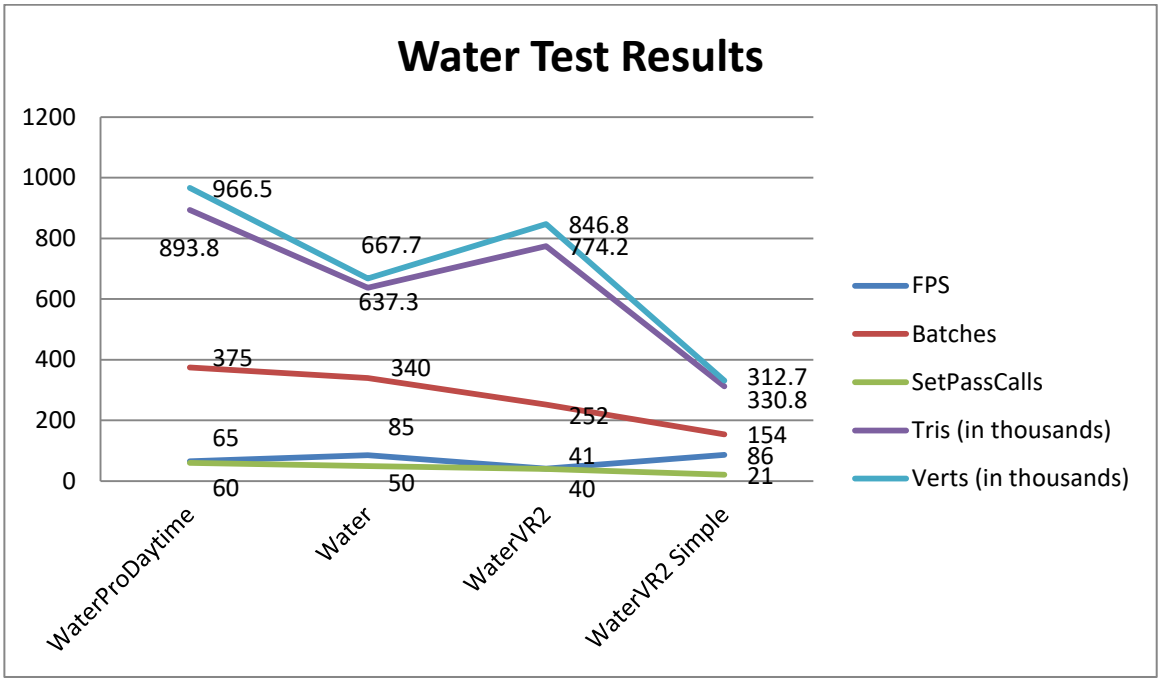


Figure 21: Water Test Results

8.10.1.4 Water4Advanced

The Moonlight River scene required a higher level water quality to give the scene the effect it need to have. The scene uses a Water4Advanced mesh from the standard asset package. The settings were changed to see what would offer the best saving in performance but still maintain the required effect.

The scene was played three different times to test the different quality settings both on computer and on the mobile device. Everything else in the scene was kept the same. The stats were then examined to see how much processing was lost due to quality change.

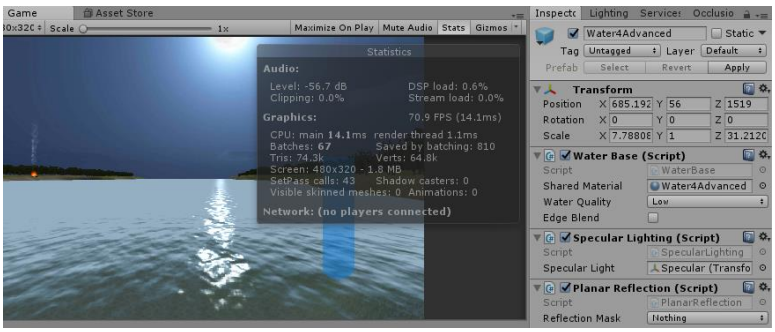


Figure 22: Water4Advanced Low Quality

With a low quality settings (see Figure 22) the water still had a realistic effect with waves and reflections. This option also had the lowest processing cost with 67 batches and 43 SetPass call. It also had the highest FPS of all the quality settings tested.

The medium quality (see Figure 23) had a nicer looking water effect with more realistic looking waves and reflections. This scene only had a small difference in FPS, batches and SetPass calls compared to the low quality settings however it's the computer had to render out over thirty thousand more triangles and thirty-five thousand more vertices. This increase in quality was

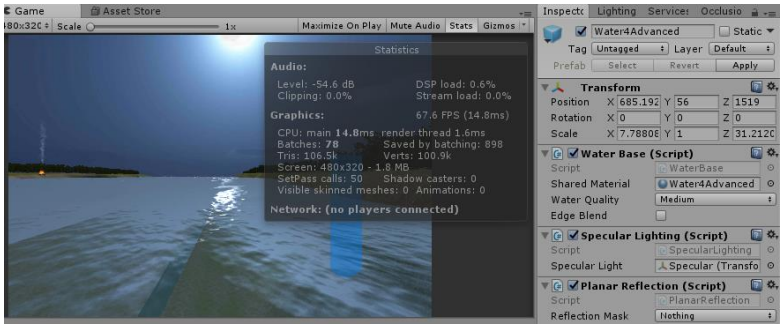


Figure 23 Water4Advanced Medium Quality

felt in the mobile test with head tracking resetting occurring more often (see Google SDK) and a slightly lower FPS.

High quality understandably had the most realistic looking water effect that was observed (see Figure 24). It featured better waves and reflections and gave a very realistic water effect. The high quality stats were similar to the medium quality stats and the same head tracking reset and FPS issues were observed in the mobile tests.

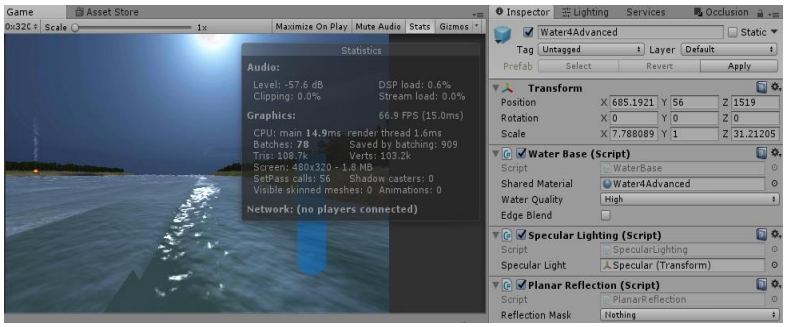


Figure 24 Water4Advanced High Quality

Due to its realistic look and lower processing needs, the low quality water was chosen for the Moonlight River scene as in the mobile tests it offered the lowest chance of head tracking reset and the highest FPS. The low quality water effect was also enough to still look realistic thus keeping the user immersed in the scene.

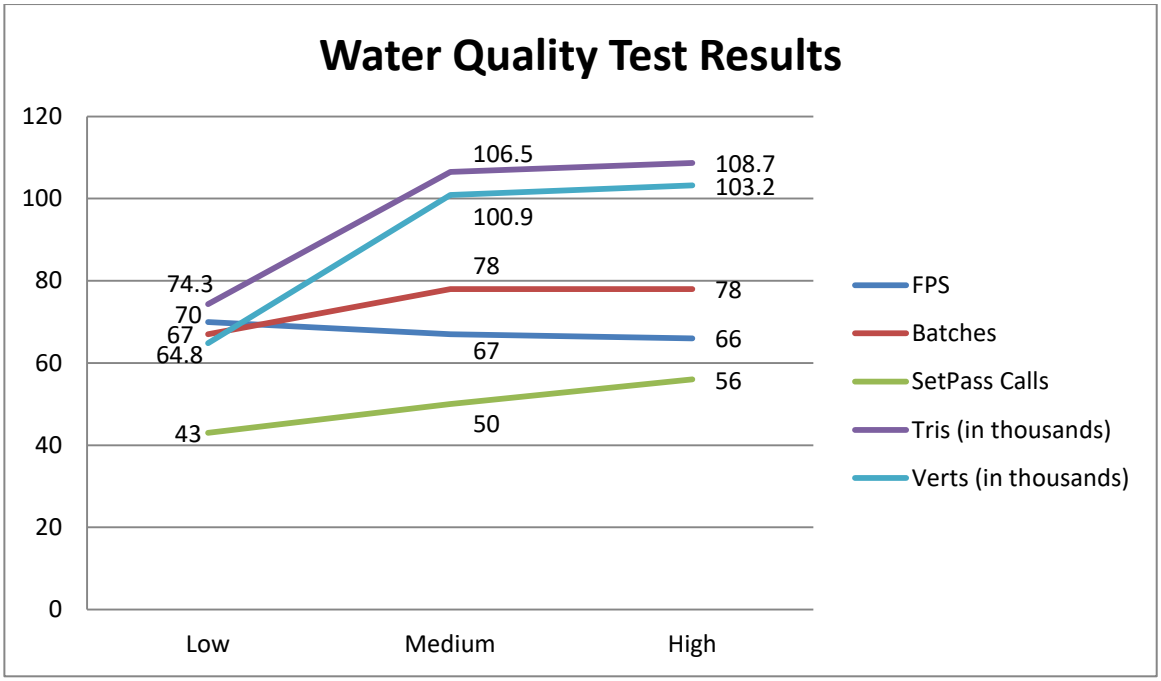


Figure 25: Water Quality Test Results

8.10.1.5 Conclusion

Once all tests were completed graphs were created (see Figure 21, Figure 25) to better visualise the results. The WaterVR2 water mesh was then chosen as it offered the highest quality with an acceptable level of performance and also the working reflections made the scene more realistic and immersive.

8.10.2 Sound

The implementation of the sounds in the prototype was important to better increase a user's immersion in the scene (Kronland-Martinet and Voinier, 2008). Once all sounds were sourced they were converted to MP3 using an online file converter (URL: https://audio.online-convert.com/convert-to-mp3), this ensured a low file size compared to WAV files making for a quicker loading game. Large sound files would affect the loading of the apps and the scene therefore

to fit in with the non-functional requirement of having the user go from app start up to scene in under 10 seconds (see Non-functional requirements).

8.10.2.1 Scene loading

To test out how fast a scene was loaded the app was started, making sure it was not running in the background of the phone, and each scene was loaded three times to get an average load time. Each scene would be loaded from the menu scene and timed on how long it took for the scene to be playing. Timings were done on an iPhone.

Scene	Mountains	Waterfall	Moonlight River
Load time 1	8.36s	8.03s	17.08s
Load time 2	6.93s	7.99s	22.89s
Load time 3	8.36s	7.56s	21.66s
Average load time	7.88s	7.86s	20.54s

Figure 26: Loading Time Results

The results showed (see Figure 26) that the loading times fit into the ten seconds loading time frame on only one of the three scenes. To lower these, the audio import setting on Unity was optimized. Every sound clip that were used in the prototype, except the UI sounds, were given the settings shown on Figure 27. By compressing the audio and lowering their quality by 50% the audio files would have a smaller size without too much noticeable loss in quality.

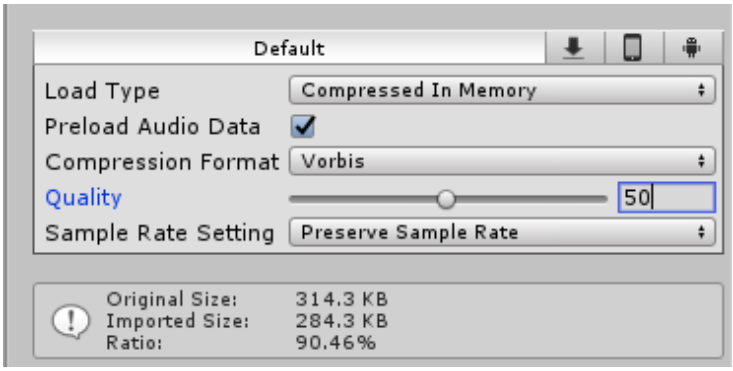


Figure 27: Unity Audio Import Settings

The timing test was then repeated to find out how much loading time was lost for each scene.

Scene	Mountains	Waterfall	Moonlight River
Load time 1	5.44s	5.12s	3.56s
Load time 2	6.45s	5.45s	3.53s
Load time 3	6.49s	5.49s	3.41s
Average load time	6.12s	5.35s	3.5s

Figure 28: Loading Times after Optimization

The results showed (see Figure 28) a significant reduction in loading times on all three scenes. Moonlight River saw the biggest drop from an average of 20.54 seconds to 3.5. This was later discovered to be due to the large file size of a water sound effect that ran throughout the whole scene which was played from an audio source attached to the player object.

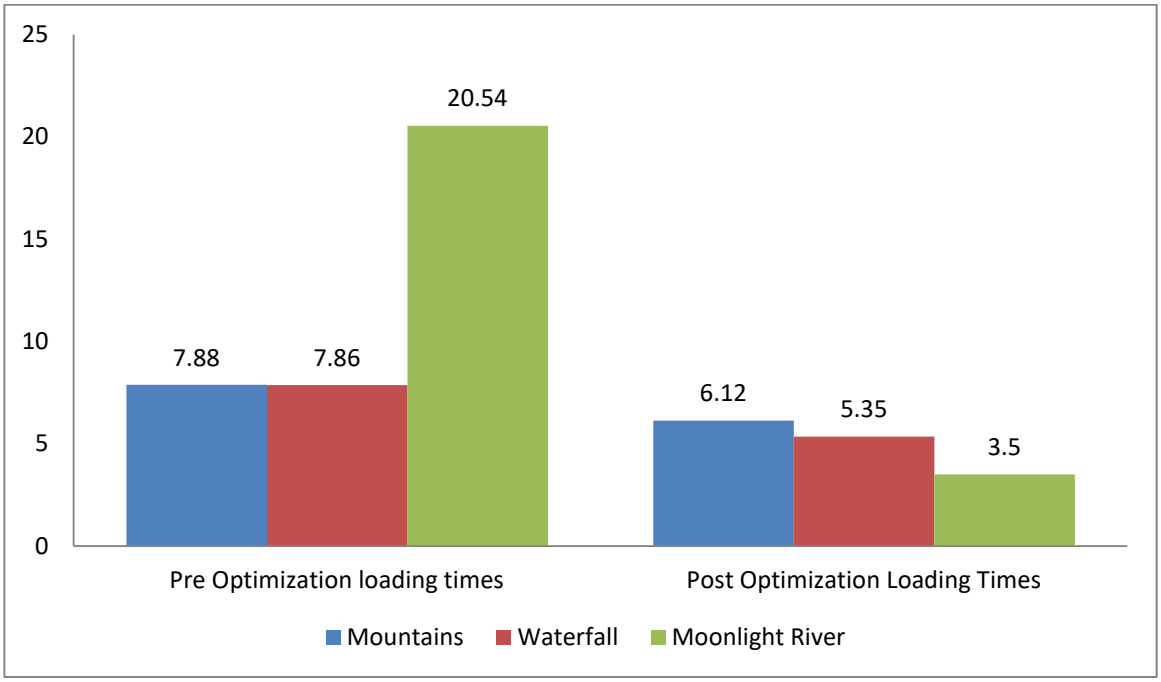


Figure 29: Sound Optimization Graph

Figure 29 shows the loading time averages in seconds of the scenes before and after the sound compression and quality settings change.

8.10.3 Quality Setting

To affect the overall performance of the game the quality settings were modified to lower the processing usage. Using an online guide (Damian Connolly, 2016) changes were made to the quality settings accordingly (see Figure 30)s.

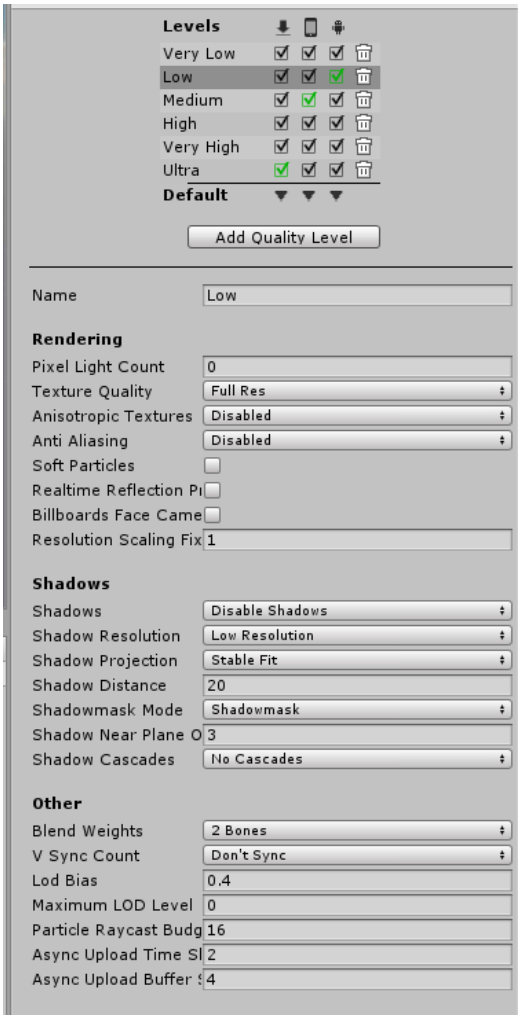


Figure 30: Quality Settings

The default quality level was set to low as most details would be lost on a small mobile screen. The pixel light count which is the maximum number of of pixel lights in the scene when forward rendering is used was reduced to zero. No scene in the prototype contained anisotropic textures so they were disabled. Anti aliasing smoths sharp edges in scenes but since this is a mobile VR app the anti alisaing effects wont have a noticable improvement in quality so this was disabled. Soft particles which uses soft blending for particles are turned off as are reflection probes as they are not used in any scene.

All shadows are disabled as any shadow on the scene has been baked in a lightmap and there are no real time light sources.

V Sync count which makes rendering be synchornised with the refresh rate of the display to avoid objects tearing on the screen is turned off as this feature wont be needed for

a mobile VR app.

Following the changes in the standard quality setting (see Figure 31), a clear increase in FPS can be seen in Figure 32 from 80-90 before to well over 210. Batches ere also lowered to 219 and SetPass calls lowered to 19. The total number of Tris and Verts on the screen saw a reduction from 600k and 646k to 106k and 62k respectivly. Mobile tests further highlighted the reduction in processing with a noticable difference in lag and problems with the VR head tracking.

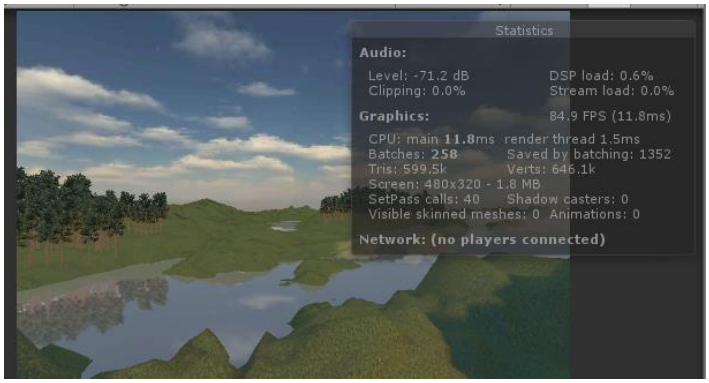


Figure 31: Before Quality Settings

The VR on the mobile tests would be slow and reset to positions without the users head movement (see 8.13 Issues,) however these problems were significantly reduced after optimization.

The graph on Figure 33 represents the before and after stats of the scene due to optimization.

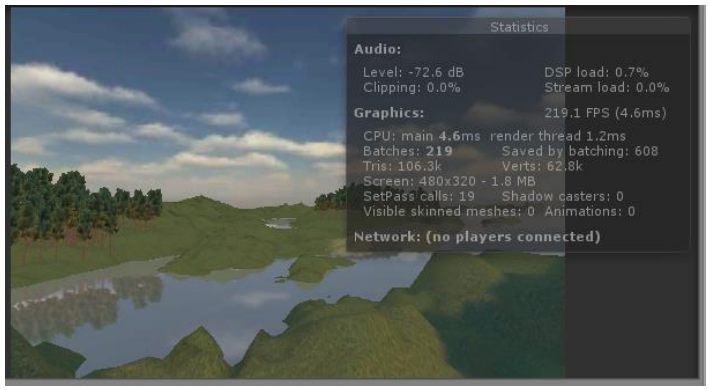


Figure 32: After Quality Settings

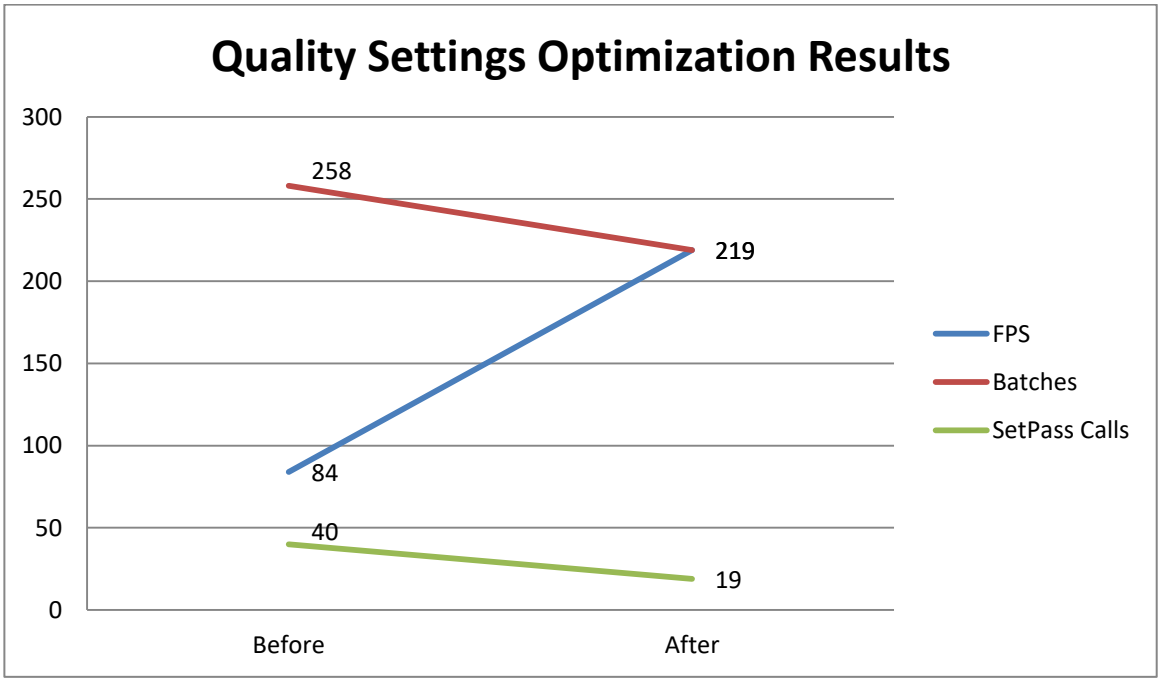


Figure 33: Quality Setting Optimization

8.11 Player Settings

This section covers the settings that were used on the player settings and why they were chosen.

The resolution and presentation settings were first looked at. Orientation settings are overridden by the VR support so to set the orientation the game will be played in the VR settings have to be disabled. Once disabled the orientation was set to auto with the allowed orientation set to landscape left and landscape right. This ensure the app would only be available to be viewed on landscape mode and would auto rotate depending which way the user was holding the phone. This was an importatnt setting as the app should not be able to be viewed in portrait mode as the menu resolution would make user selections much harder.

The Use 32-bit Display Buffer option was also ticked to make sure the display buffer could hold 32-bit colour values.

In the icon settings a simple VR Meditaion icon was created in photoshop and saved in the games images folder.

The splash image, which is always shown on any Unity game designed in the Unity personal version, was kept on default with its duration as short as possible to start to app up quicky. The VR splash image was left blank as the app is designed to start in a non-VR way before transitioning to VR when a scene is loaded.

On the other settings most things were kept on their default option. The multithreded rendering was turned on to allowed Unity to move graphics API calls from the main thread to a separate work thread. This was selected to try to improve performance and is designed for applicatios with high CPU usage.

Dynamic batching were turned on to group meshes together and let the CPU draw them in one go. Static meshes were turned on as all GameObjects in the prototype were set to static so the CPU can render them together. Both options reduce the load on the processor and make for a faster running app.

GPU skinning is an experiemntal Unity setting which is only enabled on VR builds and helps with performance and lowering processing demands from the CPU.

The VR support were enabled with two VR SDK's. the SDK's are the devices that the VR mode is going to be run on. This is set to None and Cardboard so that the splash screen and menu appear as normal with the manager script (see 8.9.1 Manager) switching to the Cardboard device when a scene is loaded. The Cardboard device also has a the transition view enabled which tells the user to put their device in a Cardboard device when the scene is first loaded.

The other settings were kept at default except for the Android TV compatibility which was set to off as it was not needed.

8.12 Build Settings

The last steps before the final build were the build settings. These would be set to keep in line with processing saving measures. Research was done to find out what are the best settings to use to provide a full working prototype for the test mobile (see Figure 34).

The build was switched to the Android platform as that is the device it would be tested and used on. All settings were kept as default. The build system was set to Gradle as this generated the APK (output package) using the Gradle build system. This kept on the default after research as this is the preferred option for exporting projects designed for Android as Gradle is the native format for Android Studio.

All scenes were then added to the build with careful consideration to what order they were in as the scene changing script works on the scene build index. Putting the scenes in a different order would cause the dropdown menu to open the wrong scene.

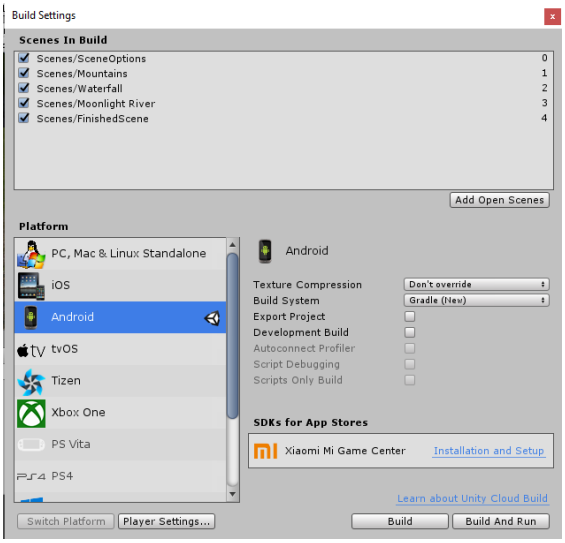


Figure 34: Build Settings

8.13 Issues

This sectioned highlights some of the issues that arose when designing the game and how and if they were overcome.

8.13.1 Build Errors

When first trying to test the scenes on a mobile device, errors kept occurring when trying to build. This wasted a lot of time as builds would normally take around 15 minutes and the error would be evident until the end. After a few days all Android SDK and Java JDK files were deleted and reinstalled which then made the builds work.

8.13.2 Google SDK

When first using the Google SDK package to the scenes a number of problems occurred. Firstly there was a lack of documentation on the website on exactly how to implement the VR settings into the game. This led to wasted time on testing the demo scenes provided by the package to figure out how it was working.

While trying to remove the Unity splash screen from appearing in VR a second Google VR SDK package was downloaded and implemented into the game following a suggestion from an online tutorial. The Google Cardboard SDK 0.6 was from a previous release dated January 12th 2016. This SDK has several issues when trying to build to an Android device. This SDK proved to be no longer compatible with Android devices and kept throwing build errors. When multiple devices were used a build was finally able to work but would crash as soon as it was opened on the phone. This SDK was the abandoned for the newer GVR SDK for Unity v1.130.1.

The other issues were related to head tracking. When the VR element was implemented into the scenes the head tracking was found to be resetting to its original position every few seconds. This led to a lost in immersion and a change in orientation which the user would have to turn their head to find again. After further research on the issue in internet forums and downloading three different releases of the Google VR SDK and testing all demo scenes, the problem was identified to be an issue with the SDK itself. This appeared to happen more often while looking at objects with high vertices and triangle counts or rapid head movements. The mobile processor and GPU were being taxed heavily and it was causing the head tracker to reset its original position in the Y axis. This error was

later dramatically reduced when the optimization settings were completed. However, the issue was not completely eliminated from the finished prototype.

The optimization settings also lowered the distance in which trees were rendered which made them slowly appear as the player moved closer. This was not ideal but it was decided to be kept as it increased FPS and decreased the head tracking resetting.

8.13.3 Water Texture

When using the standard assets water prefab within the scene, the mobile showed issues with the waters reflections. Upon further research it was discovered that the problem was with the water script attached to the prefab. The problem was fixed when an online script was found to replace the water script and give accurate reflections.

Figure 35: Reflection Error

8.13.4 Tree shadows

Trees were implemented into the terrain to make the scenes appear more realistic. Unity however failed to generate any tree shadows. Following further research a unity script was found (URL: https://bitbucket.org/snippets/sergeyo/Eq5jB7/treereplacercs-terrain-trees-converter-to) that was supposed to turn trees into objects so that light maps could be baked in with shadows. This however would crash Unity when it was run. To test the code a small terrain was made with ten trees and the code was run again. It took nearly an hour to render the shadows for the trees and with some scenes containing well over three-hundred trees this code was abandoned and to save on processing the trees would remain without shadows.

8.13.5 Post Processing

To make the scenes more realistic a post processing asset was used from the Unity asset store. Effects like motion blur higher contrast were added to further realism into the scene. However when tested all the scenes had a yellow hue to them. This was the same on multiple mobile tests. The post processing asset was therefore abandoned.

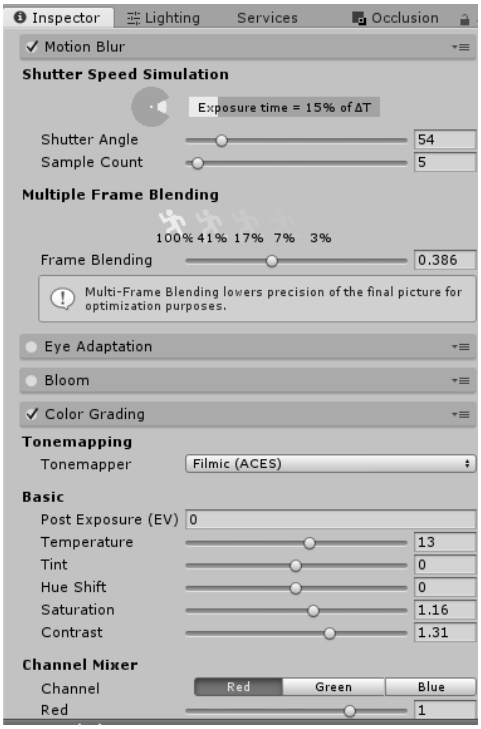


Figure 36: Post Processing Settings

9 Testing

This section will cover the testing that the prototype was put under to make sure all aspects of the app was fully functioning with all functional and non-functional requirements met.

9.1 Google Play Store

The APK of the final build of the prototype was uploaded to the Google Play store as an internal test. This allowed authorised users to download the app after receiving a link to the download page. Users were then asked to report back on the functionalities of the app to make sure that the prototype was working properly. Figure 37 shows the Google Play console release management and all released that were made.

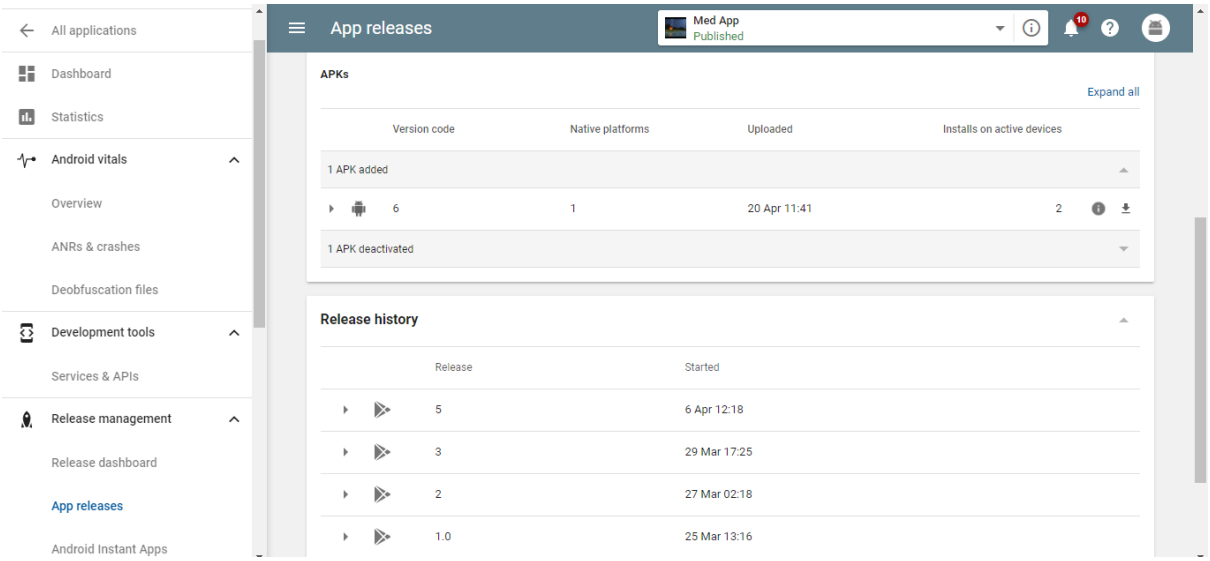


Figure 37: Google Play Console

Two different users participated on the tests. Both their email addresses were added to the download permission list and both were sent the download link. The participants were using Samsung Galaxy S6 and S9 respectively.

The Galaxy S6 test reported that the background of the menu was not rendering (see Figure 38) which could be a hardware issue or an update issue. The scene and head tracking tests worked as expected.

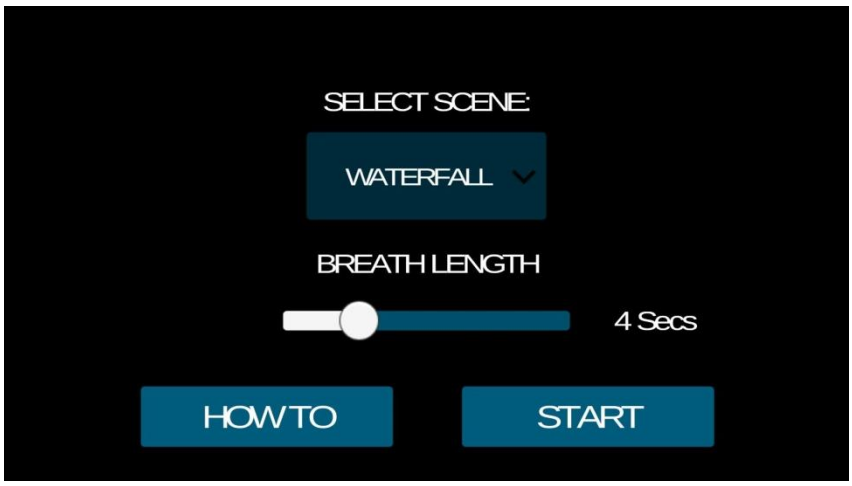


Figure 38: Menu Error

Figure 39: Waterfall Test

10 Evaluation

This section covers the testing and user evaluations that were completed after the prototype was finished. Its findings hope to check the functionality of the prototype and to answer the question of the added benefit of VR increases the engagement of the meditation application.

The prototype was evaluated using the MARS on sixteen users to assess if the level of engagement of the prototype. Users were asked to try all scenes and concentrate on following the breathing exercise on the screen. The quantifiable data was then gathered and graph and charts were created to obtain a visual understanding of the data provided.

10.1.1 MARS

The Mobile Application Rating Scale (MARS) was used to be able to assess the prototype (Stoyanov *et al.*, 2015). The quality section of the scale contains 19 questions in order to assess the app under four subscales: engagement, functionality, aesthetics and information. The quality rating is calculated using the mean score of the subscales, all of which are rated using a 5-point scale (5-Excellent, 4-Good, 3-Acceptable, 2-Poor, 1-Inadequate).

The App Classification section and Section D of the MARS scale, information, was omitted from the questions as it was not relevant to the report. The users overall satisfaction of the app are evaluated using four sections of the subjective quality section. The App-Specific questions were asked relating to using VR as an engagement tool and see if an assessment can be made if using the app has any impact on user's attitude towards VR engagement in mobile devices. Questions 3 and 4 were omitted as they were not deemed relevant.

The MARS was chosen as it provides a simple, objective, and reliable tool to assess the engagement and quality of mobile health applications.

10.1.2 Results

Table 4 shows the average result of each participant in each section of the MARS. The mean and median scores are also shown at the bottom of the table.

Figure 41 shows a pie chart based on the average scores that were given by each participant. The chart clearly shows the participants found the engagement of the app to be good on average. While a similar number of participants found it both excellent and acceptable. No participant gave a poor or inadequate rating on any of the sections of the MARS.

The functionality and aesthetics section both had all the participants give high scores above the excellent threshold so a pie chart was not needed to display their data.

Figure 42 and Figure 43 shows a bar chart of the average and mean rating on each section respectively.

Participant	Engagement	Functionality	Aesthetics
1	3.4	4.75	4.67
2	3.2	4.25	4.33
3	3.8	4.5	5
4	3.6	5	4.67
5	3.4	5	4.67
6	4	4.75	4.67
7	2.8	5	5
8	2.8	4.75	4.67
9	3.6	5	4.67
10	4	4.75	5
11	4.2	4.75	4.33
12	3.8	4.75	4.67
13	2.6	5	4.67
14	3	5	4.67
15	3.6	4.75	4.67
16	2.8	4.5	5
Mean Total	3.41	4.78	4.71
Median Total	3.5	4.75	4.67

Table 4: MARS Results

Participant	Quality	App-Specific
1	3	4.67
2	3.25	5
3	3.75	4.33
4	3	5.33
5	3.75	4.67
6	3.75	6
7	3.25	4.67
8	2.75	4.33
9	2.75	5.33
10	3.25	5
11	2.75	5.67
12	3.25	6
13	3.75	4.33
14	3.75	4.67
15	3	5.33
16	3.25	4.33
Mean Total	3.26	4.98
Median Total	3.25	4.83

Figure 40: Quality & App-Specific Results

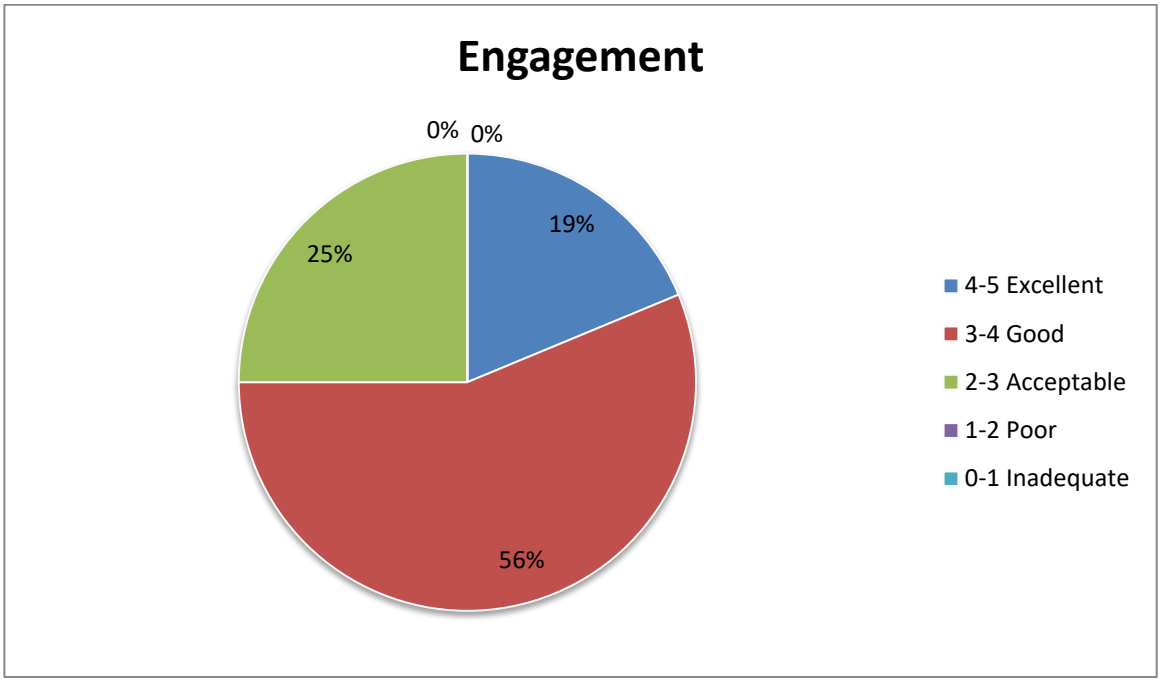


Figure 41: Engagement Evaluation

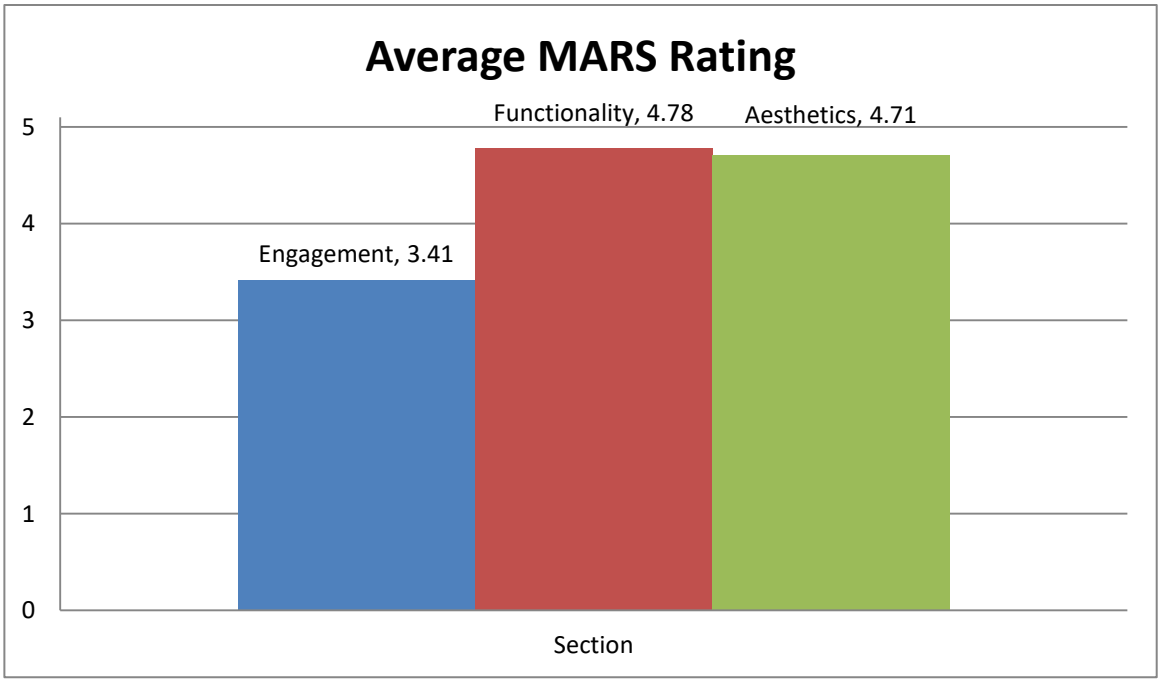


Figure 42: Average MARS Rating Chart

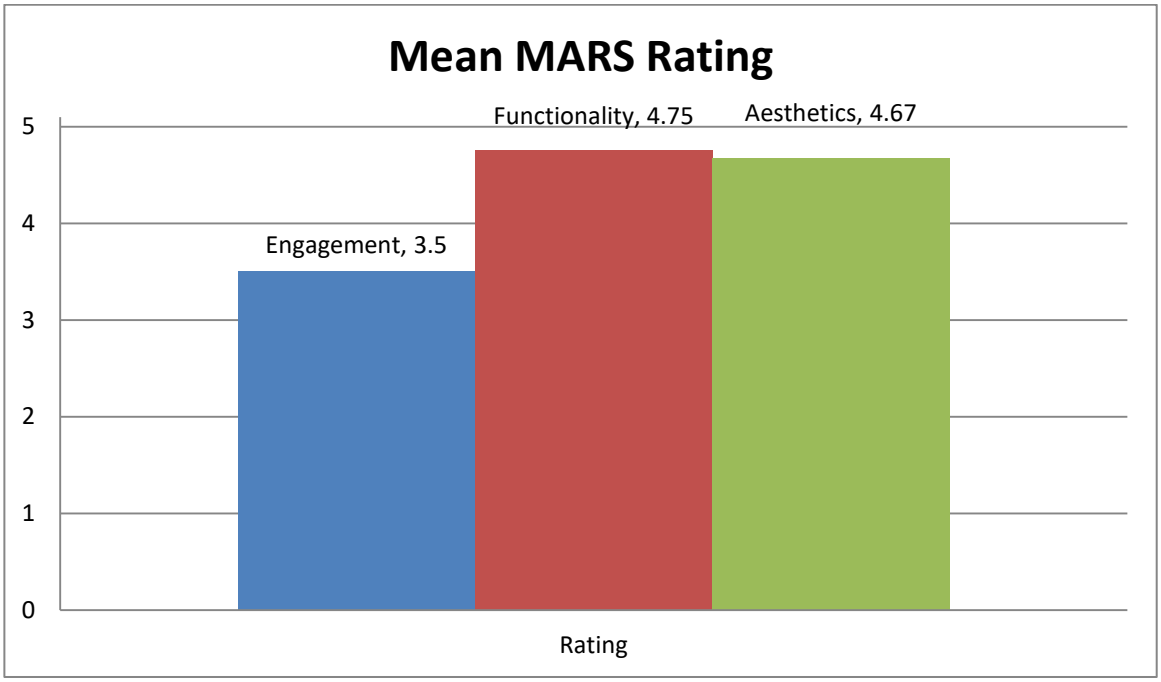
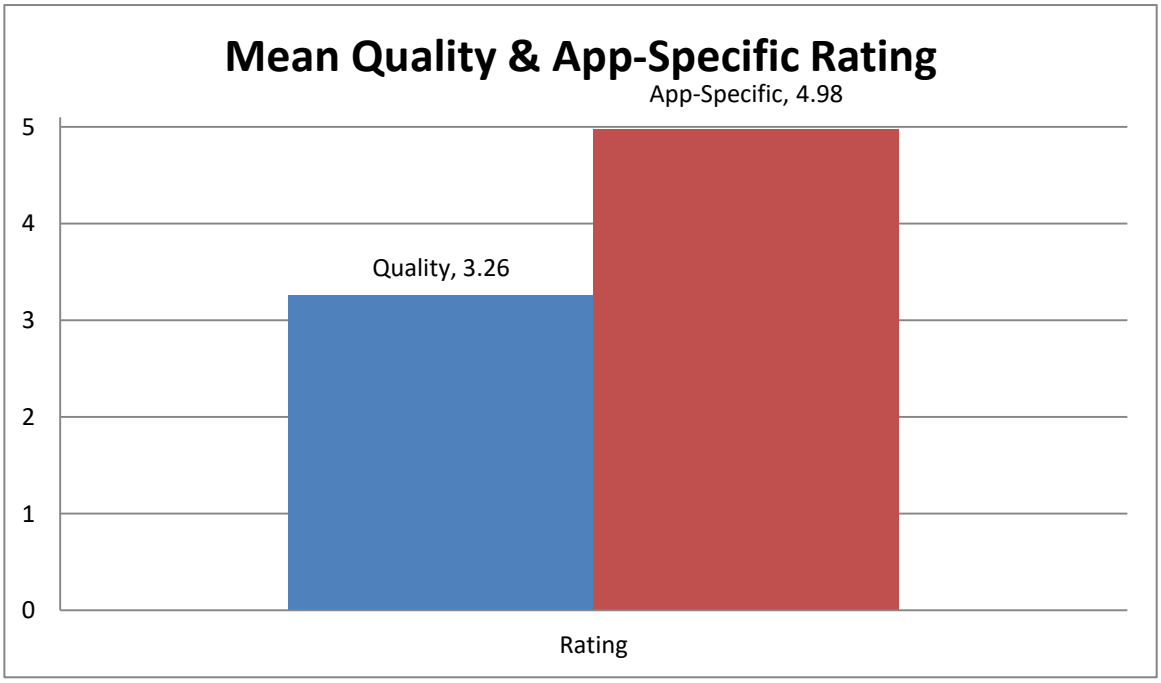
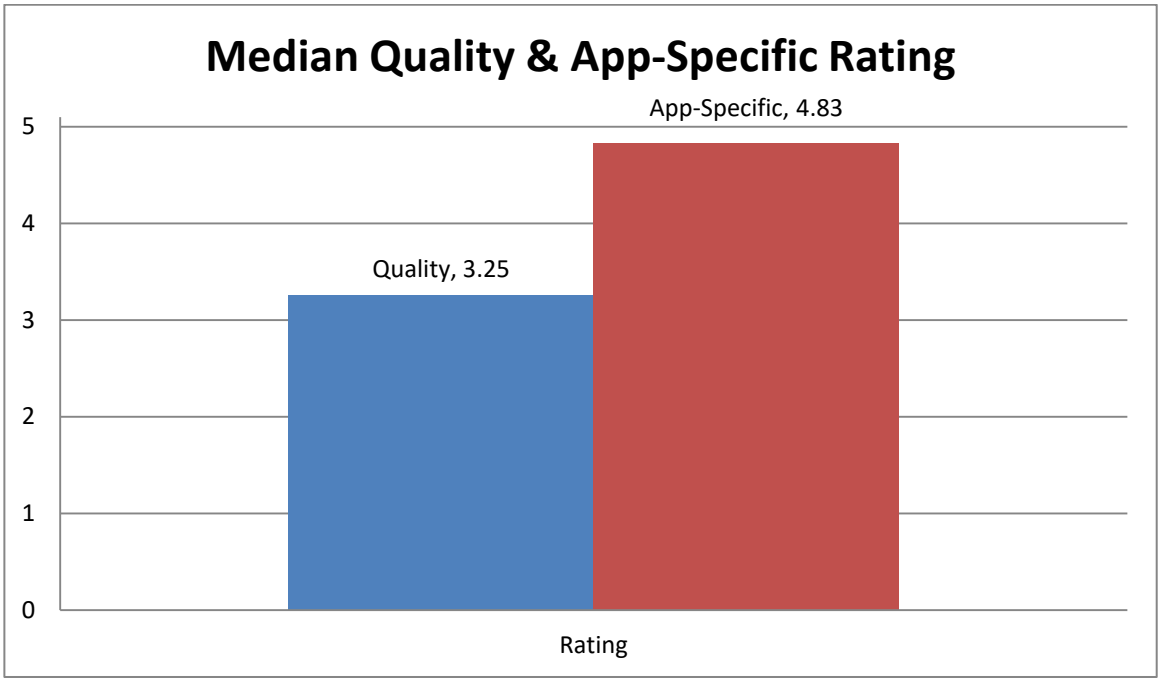


Figure 43: Mean MARS Rating Chart





10.2 Discussion

The evaluation of the engagement of the app saw an overall positive response from the participant. With a mean engagement score on the MARS of 3.5 this puts it in the good category. This along with the literature that was reviewed shows that the added feature of VR does have an increase in engagement. As meditation is something to be done regularly for it to achieve lasting positive effects it is important that a meditation app is engaging enough to have users use it regularly.

The quality and app-specific sections were used to assess the apps overall quality and the impact on user's knowledge and attitudes towards VR after using the app.

10.3 Further Questions

Extra questions directed at the participants showed that the drawbacks of VR were the use of the Google Cardboard device which can be uncomfortable and not ergonomic. Some found the scenes graphics on an average level of quality. This is due to mobile devices not having the processing power to handle high quality VR environments without issues.

The apps simple functionality and aesthetics were given excellent ratings. Upon further questions to the participants the feedback provided showed that the app was simple and easy to use due to its UI and quick start-up. The How To section also helped users on what to do to get the best experience out of the prototype.

10.4 Future Improvements

Further improvements can be made by creating more high quality scenes to further increase immersion and presence. An added feature of a tutorial scene to help the user in following the breathing guidelines can also be developed.

11 Conclusion

A range of skills and techniques were learnt in the development of this project. From designing a realistic looking environment to the implementation of VR. Since VR on mobile is a relatively new concept there was a large amount of research needed to do and online help was scarce when it came

to finding out how to fix certain problems. Time was wasted on small issues that could have been overcome with simple alternatives instead of remaining on the same problem. The final prototype met all requirements set prior to development and the evaluation found a credible link between virtual reality and engagement which made the entire project very rewarding.

The coded elements of the prototype evolved as my personal knowledge of the C# language grew. Each working script was seen as a significant milestone after a lot of trial and error. Every script worked to required level at the end thought after not giving up.

One of the main skills learnt was what it takes to develop and create an Android application. Everything from finding fonts that worked on all devices to producing a working APK was new territory.

If there was an opportunity to do the project again more time would be focused on keeping organised and adhering to the GANT chart more strictly. This would have allowed for greater user feedback and evaluation to make an ever better android application.

12 Reflection

I feel this project was carried out to the best of my capabilities. If I were to redo the prototype I would spend more time on creating bigger and more dynamic scenes. With a relatively midrange testing device I was unable to really try out higher quality scenes which would have added an increased element of immersion and made the prototype really stand out. I would have also liked to have more interesting meditation exercises to increase user's mindfulness. I am however very happy with the project and look forward to expanding on it further before doing a full beta release to the Google Play store.

13 References

Acton Smith, M. and Tew, A. (2012) 'Calm'. Calm.com Inc. Available at: https://itunes.apple.com/gb/developer/calm-com/id571800813?mt=8 (Accessed: 12 April 2018).

Bailenson, J. N. *et al.* (2008) 'The Use of Immersive Virtual Reality in the Learning Sciences: Digital Transformations of Teachers, Students, and Social Context', *Journal of the Learning Sciences*. Taylor & Francis Group , 17(1), pp. 102–141. doi: 10.1080/10508400701793141.

Baños, R. M. *et al.* (2012) 'Positive mood induction procedures for virtual environments designed for elderly people', *Interacting with Computers*. Oxford University Press, 24(3), pp. 131–138. doi: 10.1016/j.intcom.2012.04.002.

Beermann, D. and Goettel, C. (2014) 'Pacifica'.

Beijing Bluefocus E-Commerce Co. (2016) *The Impact of VR on Academic Performance*. Available at: https://cdn.uploadvr.com/wp-content/uploads/2016/11/A-Case-Study-The-Impact-of-VR-on-Academic-Performance_20161125.pdf (Accessed: 24 April 2018).

Botella, C. *et al.* (2015) 'Virtual reality exposure-based therapy for the treatment of post-traumatic stress disorder: a review of its efficacy, the adequacy of the treatment protocol, and its acceptability.', *Neuropsychiatric disease and treatment*. Dove Press, 11, pp. 2533–45. doi: 10.2147/NDT.S89542.

Bowman, D. A. and McMahan, R. P. (2007) 'Virtual Reality: How Much Immersion Is Enough?', *Computer*, 40(7), pp. 36–43. doi: 10.1109/MC.2007.257.

Buckwalter, J. G. and Rizzo, A. A. (1997) 'Virtual reality and the neuropsychological assessment of persons with neurologically based cognitive impairments.', *Studies in health technology and informatics*, 39, pp. 17–21. Available at: http://www.ncbi.nlm.nih.gov/pubmed/10168914 (Accessed: 11 April 2018).

Chan, E. A. *et al.* (2007) 'Application of a virtual reality prototype for pain relief of pediatric burn in Taiwan', *Journal of Clinical Nursing*. Wiley/Blackwell (10.1111), 16(4), pp. 786–793. doi: 10.1111/j.1365-2702.2006.01719.x.

Damian Connolly (2016) *Performance tips for Unity 2d mobile*. Available at: https://divillysausages.com/2016/01/21/performance-tips-for-unity-2d-mobile/ (Accessed: 20 April 2018).

Davis, L. *et al.* (2003) 'Enabling a continuum of virtual environment experiences', *IEEE Computer Graphics and Applications*, 23(2), pp. 10–12. doi: 10.1109/MCG.2003.1185574.

Gay, G. (2010) *Culturally responsive teaching : theory, research, and practice*. Teachers College. Available at: https://books.google.co.uk/books?id=0ZlNDwAAQBAJ (Accessed: 10 April 2018).

Goldstone, W. (2009) *Unity game development essentials*. Packt Pub. Available at: https://books.google.co.uk/books?id=WfAWzVW9IK0C&dq=unity+terrain&lr=&source=gbs_navlinks_s (Accessed: 18 April 2018).

Gonçalves, R. *et al.* (2012) 'Efficacy of Virtual Reality Exposure Therapy in the Treatment of PTSD: A Systematic Review', *PLoS ONE*. Edited by M. Slater. Public Library of Science, 7(12), p. e48469. doi: 10.1371/journal.pone.0048469.

Harris, K. and Reid, D. (2005) 'The Influence of Virtual Reality Play on Children'S Motivation', *Canadian Journal of Occupational Therapy*. SAGE PublicationsSage CA: Los Angeles, CA, 72(1), pp. 21–29. doi: 10.1177/000841740507200107.

Hua, Y. *et al.* (2015) 'The Effect of Virtual Reality Distraction on Pain Relief During Dressing Changes in

Children with Chronic Wounds on Lower Limbs', *Pain Management Nursing*. Elsevier, 16(5), pp. 685–691. doi: 10.1016/j.pmn.2015.03.001.

Kabat-Zinn, J. (2006) 'Mindfulness-Based Interventions in Context: Past, Present, and Future', *Clinical Psychology: Science and Practice*. Blackwell Publishing Ltd, 10(2), pp. 144–156. doi: 10.1093/clipsy.bpg016.

Kronland-Martinet, R. and Voinier, T. (2008) 'Real-Time Perceptual Simulation of Moving Sources: Application to the Leslie Cabinet and 3D Sound Immersion', *EURASIP Journal on Audio, Speech, and Music Processing*. Nature Publishing Group, 2008(1), pp. 1–10. doi: 10.1155/2008/849696.

Laver, K. *et al.* (2012) 'Virtual Reality for Stroke Rehabilitation', *Stroke*. American Heart Association, Inc., 43(2), pp. e20–e21. doi: 10.1161/STROKEAHA.111.642439.

Mani, M. *et al.* (2015) 'Review and Evaluation of Mindfulness-Based iPhone Apps.', *JMIR mHealth and uHealth*. JMIR Publications Inc., 3(3), p. e82. doi: 10.2196/mhealth.4328.

Matthias Koller (2016) 'Mindspace'. Available at: https://www.mindspace.org.uk/mobile-app/ (Accessed: 12 April 2018).

Mel Slater (2003) *A Note on Presence Terminology*. Available at: http://www0.cs.ucl.ac.uk/research/vr/Projects/Presencia/ConsortiumPublications/ucl_cs_papers/presence-terminology.htm (Accessed: 11 April 2018).

Merchant, Z. *et al.* (2014) 'Effectiveness of virtual reality-based instruction on students' learning outcomes in K-12 and higher education: A meta-analysis', *Computers & Education*. Pergamon, 70, pp. 29–40. doi: 10.1016/J.COMPEDU.2013.07.033.

Pacifica - About Us (2018). Available at: http://www.thinkpacifica.com/about/ (Accessed: 12 April 2018).

Parsons, T. D. *et al.* (2007) 'A Controlled Clinical Comparison of Attention Performance in Children with ADHD in a Virtual Reality Classroom Compared to Standard Neuropsychological Methods', *Child Neuropsychology*. Taylor & Francis Group , 13(4), pp. 363–381. doi: 10.1080/13825580600943473.

Péruch, P. and Gaunet, F. (1998) 'Virtual environments as a promising tool for investigating human spatial cognition', *Cahiers de Psychologie Cognitive/Current Psychology of Cognition*, 17, pp. 881–899. Available at: http://psycnet.apa.org/record/1998-12221-014 (Accessed: 24 April 2018).

'Provata Health' (2016) *Frontiers in Public Health*. Available at: https://provatahealth.com/ (Accessed: 12 April 2018).

Puddicombe, A. and Pierson, R. (2010) 'Headspace'. Available at: https://www.headspace.com/ (Accessed: 12 April 2018).

Romano, D. M. (2005) 'Virtual Reality Therapy', *Developmental Medicine & Child Neurology*. Wiley/Blackwell (10.1111), 47(9), pp. 580–580. doi: 10.1111/j.1469-8749.2005.tb01206.x.

Rothbaum, B. O. *et al.* (1995) 'Effectiveness of computer-generated (virtual reality) graded exposure in the treatment of acrophobia', 152(4), pp. 626–628. Available at: https://www.scopus.com/record/display.uri?eid=2-s2.0-0028927284&origin=inward&txGid=34b4dd3508d28d077022a1c4c4a99f30 (Accessed: 20 April 2018).

Spector, J. M. *et al.* (eds) (2014) *Handbook of Research on Educational Communications and Technology*. New York, NY: Springer New York. doi: 10.1007/978-1-4614-3185-5.

Statista (2018a) *Active virtual reality users worldwide 2014-2018*. Available at:

https://www.statista.com/statistics/426469/active-virtual-reality-users-worldwide/ (Accessed: 11 April 2018).

Statista (2018b) *Virtual reality market size worldwide by platform 2016-2020*. Available at: https://www.statista.com/statistics/528793/virtual-reality-market-size-worldwide-by-platform/ (Accessed: 24 April 2018).

Stoyanov, S. R. *et al.* (2015) 'Mobile app rating scale: a new tool for assessing the quality of health mobile apps.', *JMIR mHealth and uHealth*. JMIR Publications Inc., 3(1), p. e27. doi: 10.2196/mhealth.3422.

Tugade, M. M., Fredrickson, B. L. and Feldman Barrett, L. (2004) 'Psychological Resilience and Positive Emotional Granularity: Examining the Benefits of Positive Emotions on Coping and Health', *Journal of Personality*. Wiley/Blackwell (10.1111), 72(6), pp. 1161–1190. doi: 10.1111/j.1467-6494.2004.00294.x.

Umek, A. and Kos, A. (2016) 'Validation of smartphone gyroscopes for mobile biofeedback applications', *Personal and Ubiquitous Computing*. Springer London, 20(5), pp. 657–666. doi: 10.1007/s00779-016-0946-4.

Virvou, M. and Katsionis, G. (2008) 'On the usability and likeability of virtual reality games for education: The case of VR-ENGAGE', *Computers & Education*. Pergamon, 50(1), pp. 154–178. doi: 10.1016/J.COMPEDU.2006.04.004.

Won, A. S. *et al.* (2015) 'Two Virtual Reality Pilot Studies for the Treatment of Pediatric CRPS', *Pain Medicine*, 16(8), pp. 1644–1647. doi: 10.1111/pme.12755.

Yen, J.-C. and Lee, C.-Y. (2011) 'Exploring problem solving patterns and their impact on learning achievement in a blended learning environment', *Computers & Education*. Pergamon, 56(1), pp. 138–145. doi: 10.1016/J.COMPEDU.2010.08.012.

14 Appendices A

Mobile Application Rating Scale (MARS)

App Classification

The Classification section is used to collect descriptive and technical information about the app. Please review the app description in iTunes / Google Play to access this information.

App Name: _____

Rating this version: _____ **Rating all versions:** _____

Developer: _____

N ratings this version: _____ **N ratings all versions:** _____

Version: _____ **Last update:** _____

Cost - basic version: _____ **Cost - upgrade version:** _____

Platform: ☐☐ iPhone ☐☐ iPad ☐☐ Android

Brief description: _____

_____

Focus: what the app targets

(select all that apply)

☐☐ Increase Happiness/Well-being ☐☐ Mindfulness/Meditation/Relaxation ☐☐ Reduce negative emotions ☐☐ Depression

☐☐ Anxiety/Stress

Theoretical background/Strategies

(all that apply)

☐☐ Anger

☐☐ Behaviour Change

☐☐ Alcohol /Substance Use

☐☐ Goal Setting

☐☐ Entertainment

☐☐ Relationships

☐☐ Physical health

☐☐ Other _____

☐☐ Assessment

☐☐ Feedback

☐☐ Information/Education

☐☐ Monitoring/Tracking

☐☐ Goal setting

☐☐ Advice /Tips /Strategies /Skills training ☐☐ CBT - Behavioural (positive events) ☐☐ CBT – Cognitive (thought challenging) ☐☐ ACT - Acceptance commitment therapy ☐☐ Mindfulness/Meditation ☐☐ Relaxation

☐☐ Gratitude

☐☐ Strengths based

☐☐ Other _____

Affiliations:

☐☐ Unknown ☐☐ Commercial ☐☐ Government ☐☐ NGO ☐☐ University

Age group (all that apply)

☐☐ Children (under 12)

☐☐ Adolescents (13-17)

☐☐ Young Adults (18-25)

☐☐ Adults

☐☐ General

Technical aspects of app (all that apply)

☐☐ Allows sharing (Facebook, Twitter, etc.)

☐☐ Has an app community

☐☐ Allows password-protection

☐☐ Requires login

☐☐ Sends reminders

☐☐ Needs web access to function

App Quality Ratings

The Rating scale assesses app quality on four dimensions. All items are rated on a 5-point scale from “1.Inadequate” to “5.Excellent”. Circle the number that most accurately represents the quality of the app component you are rating. Please use the descriptors provided for each response category.

SECTION A

Engagement – fun, interesting, customisable, interactive (e.g. sends alerts, messages, reminders, feedback, enables sharing), well-targeted to audience

1. **Entertainment: Is the app fun/entertaining to use? Does it use any strategies to increase engagement through entertainment (e.g. through gamification)?**

    1 Dull, not fun or entertaining at all

    2 Mostly boring

    3 OK, fun enough to entertain user for a brief time (< 5 minutes)

    4 Moderately fun and entertaining, would entertain user for some time (5-10 minutes total)

    5 Highly entertaining and fun, would stimulate repeat use

2. **Interest: Is the app interesting to use? Does it use any strategies to increase engagement by presenting its content in an interesting way?**

    1 Not interesting at all

    2 Mostly uninteresting

    3 OK, neither interesting nor uninteresting; would engage user for a brief time (< 5 minutes)

    4 Moderately interesting; would engage user for some time (5-10 minutes total)

    5 Very interesting, would engage user in repeat use

3. **Customisation: Does it provide/retain all necessary settings/preferences for apps features (e.g. sound, content, notifications, etc.)?**

    1 Does not allow any customisation or requires setting to be input every time

    2 Allows insufficient customisation limiting functions

    3 Allows basic customisation to function adequately

    4 Allows numerous options for customisation

    5 Allows complete tailoring to the individual’s characteristics/preferences, retains all settings

4. **Interactivity: Does it allow user input, provide feedback, contain prompts (reminders, sharing options, notifications, etc.)? Note: these functions need to be customisable and not overwhelming in order to be perfect.**

    1 No interactive features and/or no response to user interaction

    2 Insufficient interactivity, or feedback, or user input options, limiting functions

    3 Basic interactive features to function adequately

    4 Offers a variety of interactive features/feedback/user input options

    5 Very high level of responsiveness through interactive features/feedback/user input options

5. **Target group: Is the app content (visual information, language, design) appropriate for your target audience?**

    1 Completely inappropriate/unclear/confusing

    2 Mostly inappropriate/unclear/confusing

    3 Acceptable but not targeted. May be inappropriate/unclear/confusing

    4 Well-targeted, with negligible issues

    5 Perfectly targeted, no issues found

A. Engagement mean score = _____

SECTION B

Functionality – app functioning, easy to learn, navigation, flow logic, and gestural design of app

6. **Performance: How accurately/fast do the app features (functions) and components (buttons/menus) work?**

    1 App is broken; no/insufficient/inaccurate response (e.g. crashes/bugs/broken features, etc.)

    2 Some functions work, but lagging or contains major technical problems

    3 App works overall. Some technical problems need fixing/Slow at times

    4 Mostly functional with minor/negligible problems

    5 Perfect/timely response; no technical bugs found/contains a 'loading time left' indicator

7. **Ease of use: How easy is it to learn how to use the app; how clear are the menu labels/icons and instructions?**

    1 No/limited instructions; menu labels/icons are confusing; complicated

    2 Useable after a lot of time/effort

    3 Useable after some time/effort

    4 Easy to learn how to use the app (or has clear instructions)

    5 Able to use app immediately; intuitive; simple

8. **Navigation: Is moving between screens logical/accurate/appropriate/ uninterrupted; are all necessary screen links present?**

    1 Different sections within the app seem logically disconnected and random/confusing/navigation is difficult

    2 Usable after a lot of time/effort

    3 Usable after some time/effort

    4 Easy to use or missing a negligible link

    5 Perfectly logical, easy, clear and intuitive screen flow throughout, or offers shortcuts

9. **Gestural design: Are interactions (taps/swipes/pinches/scrolls) consistent and intuitive across all components/screens?**

    1 Completely inconsistent/confusing

    2 Often inconsistent/confusing

    3 OK with some inconsistencies/confusing elements

    4 Mostly consistent/intuitive with negligible problems

5 Perfectly consistent and intuitive

B. Functionality mean score = _____

SECTION C

Aesthetics – graphic design, overall visual appeal, colour scheme, and stylistic consistency

10. **Layout: Is arrangement and size of buttons/icons/menus/content on the screen appropriate or zoomable if needed?**
    1 Very bad design, cluttered, some options impossible to select/locate/see/read device display not optimised
    2 Bad design, random, unclear, some options difficult to select/locate/see/read
    3 Satisfactory, few problems with selecting/locating/seeing/reading items or with minor screen-size problems
    4 Mostly clear, able to select/locate/see/read items
    5 Professional, simple, clear, orderly, logically organised, device display optimised. Every design component has a purpose

11. **Graphics: How high is the quality/resolution of graphics used for buttons/icons/menus/content?**

    1 Graphics appear amateur, very poor visual design - disproportionate, completely stylistically inconsistent
    2 Low quality/low resolution graphics; low quality visual design – disproportionate, stylistically inconsistent
    3 Moderate quality graphics and visual design (generally consistent in style)
    4 High quality/resolution graphics and visual design – mostly proportionate, stylistically consistent
    5 Very high quality/resolution graphics and visual design - proportionate, stylistically consistent throughout

12. **Visual appeal: How good does the app look?**

    1 No visual appeal, unpleasant to look at, poorly designed, clashing/mismatched colours
    2 Little visual appeal – poorly designed, bad use of colour, visually boring
    3 Some visual appeal – average, neither pleasant, nor unpleasant
    4 High level of visual appeal – seamless graphics – consistent and professionally designed
    5 As above + very attractive, memorable, stands out; use of colour enhances app features/menus

C. Aesthetics mean score = _____

SECTION D

Information – Contains high quality information (e.g. text, feedback, measures, references) from a credible source. Select N/A if the app component is irrelevant.

13. **Accuracy of app description (in app store): Does app contain what is described?**

    1 Misleading. App does not contain the described components/functions. Or has no description
    2 Inaccurate. App contains very few of the described components/functions
    3 OK. App contains some of the described components/functions
    4 Accurate. App contains most of the described components/functions
    5 Highly accurate description of the app components/functions

14. **Goals: Does app have specific, measurable and achievable goals (specified in app store description or within the app itself)?**

N/A Description does not list goals, or app goals are irrelevant to research goal (e.g. using a game for educational purposes)

1 App has no chance of achieving its stated goals

2 Description lists some goals, but app has very little chance of achieving them

3 OK. App has clear goals, which may be achievable.

4 App has clearly specified goals, which are measurable and achievable

5 App has specific and measurable goals, which are highly likely to be achieved

15. Quality of information: Is app content correct, well written, and relevant to the goal/topic of the app?

N/A There is no information within the app

1 Irrelevant**/**inappropriate/incoherent/incorrect

2 Poor. Barely relevant/appropriate/coherent**/**may be incorrect

3 Moderately relevant/appropriate/coherent**/**and appears correct

4 Relevant**/**appropriate/coherent/correct

5 Highly relevant, appropriate, coherent, and correct

16. **Quantity of information: Is the extent coverage within the scope of the app; and comprehensive but concise?**

N/A There is no information within the app

1 Minimal or overwhelming

2 Insufficient or possibly overwhelming

3 OK but not comprehensive or concise

4 Offers a broad range of information, has some gaps or unnecessary detail; or has no links to more information and resources

5 Comprehensive and concise; contains links to more information and resources

17. **Visual information: Is visual explanation of concepts – through charts/graphs/images/videos, etc. – clear, logical, correct?**

N/A There is no visual information within the app (e.g. it only contains audio, or text)

1 Completely unclear/confusing/wrong or necessary but missing

2 Mostly unclear/confusing/wrong

3 OK but often unclear/confusing/wrong

4 Mostly clear/logical/correct with negligible issues

5 Perfectly clear/logical/correct

18. **Credibility: Does the app come from a legitimate source (specified in app store description or within the app itself)?**

1 Source identified but legitimacy/trustworthiness of source is questionable (e.g. commercial business with vested interest)

2 Appears to come from a legitimate source, but it cannot be verified (e.g. has no webpage)

3 Developed by small NGO/institution (hospital/centre, etc.) /specialised commercial business, funding body

4 Developed by government, university or as above but larger in scale

5 Developed using nationally competitive government or research funding (e.g. Australian Research Council, NHMRC)

19. **Evidence base: Has the app been trialled/tested; must be verified by evidence (in published scientific literature)?**

N/A The app has not been trialled/tested

1 The evidence suggests the app does not work

2 App has been trialled (e.g., acceptability, usability, satisfaction ratings) and has partially positive outcomes in studies that are not randomised controlled trials (RCTs), or there is little or no contradictory evidence.

3 App has been trialled (e.g., acceptability, usability, satisfaction ratings) and has positive outcomes in studies that are not RCTs, and there is no contradictory evidence.

4 App has been trialled and outcome tested in 1-2 RCTs indicating positive results

5 App has been trialled and outcome tested in > 3 high quality RCTs indicating positive results

D. Information mean score = _____ *

* Exclude questions rated as “N/A” from the mean score calculation.

App subjective quality

SECTION E

---

20. Would you recommend this app to people who might benefit from it?

1	**Not at all**	I would not recommend this app to anyone
2		There are very few people I would recommend this app to
3	**Maybe**	There are several people whom I would recommend it to
4		There are many people I would recommend this app to
5	**Definitely**	I would recommend this app to everyone

21. How many times do you think you would use this app in the next 12 months if it was relevant to you?

1 **None**

2 1-2

3 3-10

4 10-50

5 >50

22. Would you pay for this app?

1 No

3 Maybe

5 Yes

23. What is your overall star rating of the app?

1	★★	One of the worst apps I've used
2	★★★★	
3	★★★★★★	Average

4 ★★★★★★★★

5 ★★★★★★★★★★One of the best apps I've used

Scoring

App quality scores for

SECTION

A: Engagement Mean Score = _____

B: Functionality Mean Score = _____

C: Aesthetics Mean Score = _____

D: Information Mean Score = _____

App quality mean Score = _____

App subjective quality Score = _____

App-specific

These added items can be adjusted and used to assess the perceived impact of the app on the user's knowledge, attitudes, intentions to change as well as the likelihood of actual change in the target health behaviour.

SECTION F

---

1. **Awareness: This app is likely to increase awareness of the importance of addressing [insert target health behaviour]**

Strongly disagree				Strongly Agree
1	2	3	4	5

2. **Knowledge: This app is likely to increase knowledge/understanding of [insert target health behaviour]**

Strongly disagree				Strongly Agree
1	2	3	4	5

3. **Attitudes: This app is likely to change attitudes toward improving [insert target health behaviour]**

Strongly disagree				Strongly Agree
1	2	3	4	5

4. **Intention to change: This app is likely to increase intentions/motivation to address [insert target health behaviour]**

Strongly disagree				Strongly Agree
1	2	3	4	5

5. **Help seeking: Use of this app is likely to encourage further help seeking for [insert target health behaviour] (if it's required)**

Strongly disagree				Strongly Agree
1	2	3	4	5

6. Behaviour change: Use of this app is likely increase/decrease [insert target health behaviour]

Strongly disagree				Strongly Agree
1	2	3	4	5

15 Appendices B

15.1 Manager

```
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;
using UnityEngine.VR;
using System.Collections;

public class Manager : MonoBehaviour
{
    //sets up all UI public variables to be dragged in
    [Header("UI Elements")]
    public Button submitButton;
    public Button howToButton;
    public Button backButton;
    public GameObject howToUI;
    public Dropdown sceneSelect;
    public Slider breathSlider;
    public Image loadBar;
    public GameObject UICanvas;

    int sceneSelection;
    private float breathLength;
    public float breathOption;
    public static Manager instance = null;


    void Awake()
    {

        //Check if instance already exists
        if (instance == null)
        {

            //if not, set instance to this
            instance = this;
        }

        //If instance already exists and it's not this:
        else if (instance != this)
        {
            //Then destroy this. This enforces our singleton pattern, meaning there
can only ever be one instance of a GameManager.
            Destroy(gameObject);
        }

        //Sets this to not be destroyed when reloading scenedestor
        DontDestroyOnLoad(gameObject);
    }


    public void Start()
    {
        //make sure VR settings are disabled for the menu screen upon start
        VRSettings.enabled = false;

        //add functions to the UI buttons
        submitButton.onClick.AddListener(() => TaskOnClick());
        howToButton.onClick.AddListener(() => HowToMenu());
        backButton.onClick.AddListener(() => HowToMenuBack());
```

```
    }

    //function to run when button has been pressed
    public void TaskOnClick()
        {
        //finding the user options on scene and breath length
        sceneSelection = sceneSelect.value;
        breathLength = breathSlider.value;

        //start coroutine to play selected scene
        StartCoroutine(LoadScene(sceneSelection + 1));
    }

    //function to set the How To screen active
    public void HowToMenu()
    {
        howToUI.SetActive(true);
    }

    //function to disable the How To screen
    public void HowToMenuBack ()
    {
        howToUI.SetActive(false);
    }

    //coroutine to load the next scene and use the loading bar
    IEnumerator LoadScene(int sceneIndex)
    {
        //load the chosen scene and work out its progress to be set to
        //the fill amount of the loadbar
        AsyncOperation nextScene = SceneManager.LoadSceneAsync(sceneIndex);

        while (!nextScene.isDone)
        {
            float progress = Mathf.Clamp01(nextScene.progress / .9f);
            loadBar.fillAmount = nextScene.progress;
            yield return null;
        }
        //start the loadDevice coroutine to set the VR device to google cardboard
        StartCoroutine(LoadDevice("cardboard"));
        yield return null;
    }

    void Update()
    {
        //set the user selected breath option to the public breathLength variable
        breathOption = breathLength;

        //check if the scene is not the menu scene
        //if so disable the menu GUI
        if (SceneManager.GetActiveScene().buildIndex != 0)
        {
            loadBar.fillAmount = 0;
            UICanvas.SetActive(false);
        } else
        {
            UICanvas.SetActive(true);
        }
            if (Input.GetKeyDown(KeyCode.Escape))
        {
            //uses the VR close icon to close the current scene and open the menu
scene
```

```
            Fader(1);
            StartCoroutine(SceneOptions(2f));
        }
    }

    //public function to reset the scene when meditation has finished
    public void Reset()
    {
        StartCoroutine(FinishScene(5));
    }

    //coroutine to set the VR device and enable the VR settings
    IEnumerator LoadDevice(string newDevice)
    {
        VRSettings.LoadDeviceByName(newDevice);
        yield return null;
        VRSettings.enabled = true;
    }

    //coroutine to load the finish scene, wait a few seconds then load the
    //menu scene
    IEnumerator FinishScene(float delay)
    {
        SceneManager.LoadScene(4);
        yield return new WaitForSeconds(delay);
        StartCoroutine(SceneOptions(5));
    }

    //function to use to fade script
    public void Fader(int fade)
    {
        GetComponent<Fading>().BeginFade(fade);
    }

    // called when level is loaded to fade the screen
    void OnLevelWasLoaded()
    {
        Fader(-1);
    }

    //coroutine to set the VR device to none and disable to VR setting
    //wait a few seconds before the menu scene is loaded
    IEnumerator SceneOptions(float delay) {
        VRSettings.LoadDeviceByName("None");
        VRSettings.enabled = false;
        yield return new WaitForSeconds(delay);
        SceneManager.LoadScene(0);
        yield break;
    }

    //reload menu scene when user comes off the application
    /*void OnApplicationPause(bool pause)
    {
        StartCoroutine(SceneOptions(2));
    }*/
    }
```

15.2 AudioPlay

```
using UnityEngine;
```

```
public class AudioPlay : MonoBehaviour
{

    public AudioSounds[] sounds;
    private Vector3[] positionArray;
    private float triggerDistance;

    AudioSource playAudio;
    float proximity = 0f;

    public static AudioPlay instance = null;

    // Use this for initialization
    void Awake()
    {
        //adds Audio Sounds script attributes to the inspector
        foreach (AudioSounds s in sounds)
        {
            s.source = s.soundSource.AddComponent<AudioSource>();
            s.source.clip = s.clip;

            s.source.volume = s.volume;
            s.source.pitch = s.pitch;
            s.source.loop = s.loop;
            s.source.spatialBlend = s.spacialBlend;
            s.source.maxDistance = s.maxDistance;

            //chosen distance will be distace the audio starts working in and the max
distance of the audio source
            triggerDistance = s.triggerDistance;
        }
    }

    void Start()
    {

        //Array to hold audio sources Vector 3 positions in relation to the audio
gameobjects
        positionArray = new Vector3[sounds.Length];

        for (int i = 0; i < sounds.Length; i++)
        {
            Vector3 position = sounds[i].source.transform.position;
            positionArray[i] = position;
        }

    }


    // Update is called once per frame
    void Update()
    {
            //checks if player is under the distance so that the audio can start
playing
            for (int i = 0; i < sounds.Length; i++)
            {
                playAudio = sounds[i].soundSource.GetComponent<AudioSource>();
                proximity = Vector3.Distance(transform.position, positionArray[i]);
                triggerDistance = sounds[i].triggerDistance;

                if (proximity < triggerDistance)
                {
```

```
                    //checks that the audio is not already playing
                    if (!playAudio.isPlaying)
                    {
                        playAudio.PlayOneShot(playAudio.clip);
                    }
                }
                else
                {
                    //stops audio when ditance is above the chosen distance
                    playAudio.Stop();
                }
            }
    }
}
```

15.3 AudioSounds

```
using UnityEngine.Audio;
using UnityEngine;

[System.Serializable]
public class AudioSounds {

    //creates public variables for the audio play script to use
    public GameObject soundSource;

    //public variable for the sound clips
    public AudioClip clip;

    //public variables to set sound volume, pitch and spacial blend for 3D sound
    [Range(0f, 1f)]
    public float volume;
    [Range(.1f, 3f)]
    public float pitch;
    [Range(0f, 1f)]
    public float spacialBlend;

    //public bool to set the clip to loop
    public bool loop;

    //public float to set the max distance the sound can be heard from
    public float maxDistance;

    //public float to set the distance the sound will start playing
    public float triggerDistance;

    //hidden public audio source to find the objects audio source or create one
    [HideInInspector]
    public AudioSource source;


}
```

15.4 CameraMovement

```
using System.Collections;
using UnityEngine;

public class CameraMovement : MonoBehaviour {

    //set the cameras default speed
```

```
    public static float defaultSpeed = 10f;

    //float to set the speed of the camera
    public float speed;

    //targets transform used to check proximity
    private Transform target;
    //set the index of the waypoint and public float to change the distance
    //in which the next waypoint is triggered
    private int waypointIndex = 0;
    public float wpDistance = 10f;

       // Use this for initialization
       void Start () {
        //set all targets transform to an array
        target = Waypoints.points[0];
        //set the speed to equals the default speed
        speed = defaultSpeed;
       }

       // Update is called once per frame
       void Update () {
        //check if the targets position compared to the cameras
        Vector3 dir = target.position - transform.position;
        //tranform the camera position to go towards the next waypoint using the speed
        //variable and Time.deltaTime
        transform.Translate(dir.normalized * speed * Time.deltaTime, Space.World);

        //check distance to next waypoint
        if (Vector3.Distance(transform.position, target.position) <= wpDistance)
        {
            //if distance to next way point is bellow the chosen distance
            //the GetNextWaypoint function starts
            GetNextWaypoint();
        }
       }


    void GetNextWaypoint()
    {
        //checks if the next waypoint is second to last on the array
        if (waypointIndex >= Waypoints.points.Length - 2)
        {
            //if waypoint is second to last then fade out the scene and start the
finishScene coroutine
            Manager.instance.Fader(1);
            StartCoroutine(FinishScene(2));
        }

        //adds a value to the waypoint index and changes the camera to follow the next
waypoint
        waypointIndex++;
        target = Waypoints.points[waypointIndex];
    }

    IEnumerator FinishScene(int delay)
    {
        //if the scene is finished fade out the breathing sounds and wait a few
seconds
        //before starting the Reset function on the managers script
        GameObject.Find("BreathBar").GetComponent<AudioSource>().volume -=
Time.deltaTime /3;
```

```
        yield return new WaitForSeconds(delay);
        Manager.instance.Reset();
    }


}
```

15.5 DontDestroy

```
using UnityEngine;

public class DontDestroy : MonoBehaviour {

    public static DontDestroy instance = null;

    void Awake()
    {
        //Check if instance already exists
        if (instance == null)
        {

            //if not, set instance to this
            instance = this;
        }

        //If instance already exists and it's not this:
        else if (instance != this)
        {
            //Then destroy this. This enforces our singleton pattern, meaning there
can only ever be one instance of a GameManager.
            Destroy(gameObject);
        }

        //Sets this to not be destroyed when reloading scene
        DontDestroyOnLoad(gameObject);
    }
}
```

15.6 Fading

```
using UnityEngine;
using System.Collections;

public class Fading : MonoBehaviour {

    //set the fading texture and fade speed
       public Texture2D fadingTexture;
       public float fadingSpeed = 0.9f;

       private int drawDepth = -1000;
       private float alpha = 1.0f;
       private int fadeDir = -1;

       void OnGUI()
       {
        // fade in and out the alpha using direction in seconds
              alpha += fadeDir * fadingSpeed * Time.deltaTime;
        // clamp the fade between 0 and 1
              alpha = Mathf.Clamp01(alpha);

        // set the colour of the GUI
```

```
            GUI.color = new Color (GUI.color.r, GUI.color.g, GUI.color.b, alpha);
        // render the black texture on top
            GUI.depth = drawDepth;
        // draw the texture to fit the screen
            GUI.DrawTexture(new Rect(0, 0, Screen.width, Screen.height),
fadingTexture);
       }

    // set the fadeDir to direction of parameter making scene fade in
       public float BeginFade (int direction)
       {
            fadeDir = direction;
            return (fadingSpeed);
       }


}
```

15.7 MeditationUI

```
using UnityEngine;
using UnityEngine.UI;

public class MeditationUI : MonoBehaviour {

    Image fillImg;
    float timer;
    float interval;

    public AudioClip inhale;
    public AudioClip exhale;

    void Awake()
    {
        //finds the manager object which is storing the breath option script
        Manager manager  = GameObject.Find("_Manager").GetComponent<Manager>();

        //interval is changed to match user chosen breath length
        interval = manager.breathOption;
    }

    // Use this for initialization
    void Start () {
        fillImg = GetComponent<Image>();
    }

       // Update is called once per frame
       void Update () {
        //sets the timer to bounce between 0 and 1 based on the user
        //chosen interval then change the breath bars fill amount accordingly
        timer = Mathf.PingPong(Time.timeSinceLevelLoad / interval, 1);
        fillImg.fillAmount = timer;

        //when the fill amount is at the bottom play the inhale sound clip
        //when it is at the top play in exhale sound clip
        if (timer <= .1f && !GetComponent<AudioSource>().isPlaying)
        {
            gameObject.GetComponent<AudioSource>().PlayOneShot(inhale, 1f);
        } else if (timer >= .9f && !GetComponent<AudioSource>().isPlaying)
        {
            gameObject.GetComponent<AudioSource>().PlayOneShot(exhale, 1f);
```

```
        }
    }
}
```

15.8 SliderVal

```
using UnityEngine;
using UnityEngine.UI;

public class SliderVal : MonoBehaviour {

    Text txt;

       // Update is called once per frame
       void Update () {

        //checks slider value and displays it on the text box next to it
        txt = GetComponent<Text>();
        float breathSliderVal =
GameObject.FindGameObjectWithTag("BreathOption").GetComponent<Slider>().value;
        txt.text = breathSliderVal + " Secs";
    }
}
```

15.9 Timer

```
using System.Collections;
using UnityEngine;
using UnityEngine.UI;

public class Timer : MonoBehaviour {
    Image fillImg;
    public float timeAmt = 10;
    float time;

       // Use this for initialization
       void Start () {
        fillImg = this.GetComponent<Image>();
        time = timeAmt;
       }

       // Update is called once per frame
       void Update () {
        if(time > 0)
        {
            time -= Time.deltaTime;
            fillImg.fillAmount = time / timeAmt;

            if (time < 0)
            {
                fillImg.fillAmount = 1;
                time = timeAmt;
            }
        }

        Debug.Log(time);
       }
}
```

15.10 Waypoints

```
using UnityEngine;

public class Waypoints : MonoBehaviour {

    public static Transform[] points;

       void Awake () {
        points = new Transform[transform.childCount];
        for (int i = 0; i < points.Length; i++)
        {
            points[i] = transform.GetChild(i);
        }
       }


}
```

16 Appendices C

COMP[1682] Project Proposal

Increasing Engagement in Meditation Apps with the Use of Virtual Reality

Will Oliveira

Computer Games Design & Development

731609

1 Overview

The emerging market of guided meditation apps is not a new one. Evidence has shown that a link between student well-being and academic outcomes. Studies have also shown the effectiveness of meditation and how guided meditation apps are an effective way to treat a range of different mental health issues. For this reason, this report will show how using virtual reality technology and immersive therapy can be used to help increase engagement in guided meditation apps. The report will explore research into stress relief exercises which will be modelled inside a 3D virtual reality space and can be used by anyone with a modern smartphone and a Google Cardboard device.

This report will aim discover if virtual reality can improve increase the use of meditation apps and if so by how much. The report will examine how different scenarios and exercises set in a virtual reality can increase the engagement of it by users.

The participants will be immersed into a virtual reality scenario and then a simple questionnaire will help us evaluate how effective the experience was and if they are more likely to use it.

Keywords: virtual, reality, stress, anxiety, immersive, therapy, engagement

2 Aim

To evaluate whether a virtual reality meditation app gives increased engagement to its non-VR counterpart

3 Objectives

[] = allocated time in hours

3.1 Research

3.1.1 Research into virtual reality [25]

3.1.2 Research into methods used to relieve stress using virtual reality [30]

3.1.3 Research into existing virtual reality therapy [25]

3.2 Development

3.2.1 Development of a 3D environment using 3DS Max using research based design [30]

3.2.2 Integration of 3D environment into Unity game engine [30]

3.2.3 Incorporating virtual reality support on the 3D environment [25]

3.3 Implementation

3.3.1 Using researched based methods to show users breathing and relaxing exercises [25]

3.3.2 Users shown working prototype [10]

3.4 Evaluation

3.4.1 Users will be asked to fill in a questionnaire before and after using the product [10]

3.4.2 Users engagement levels will be compared and evaluated [10]

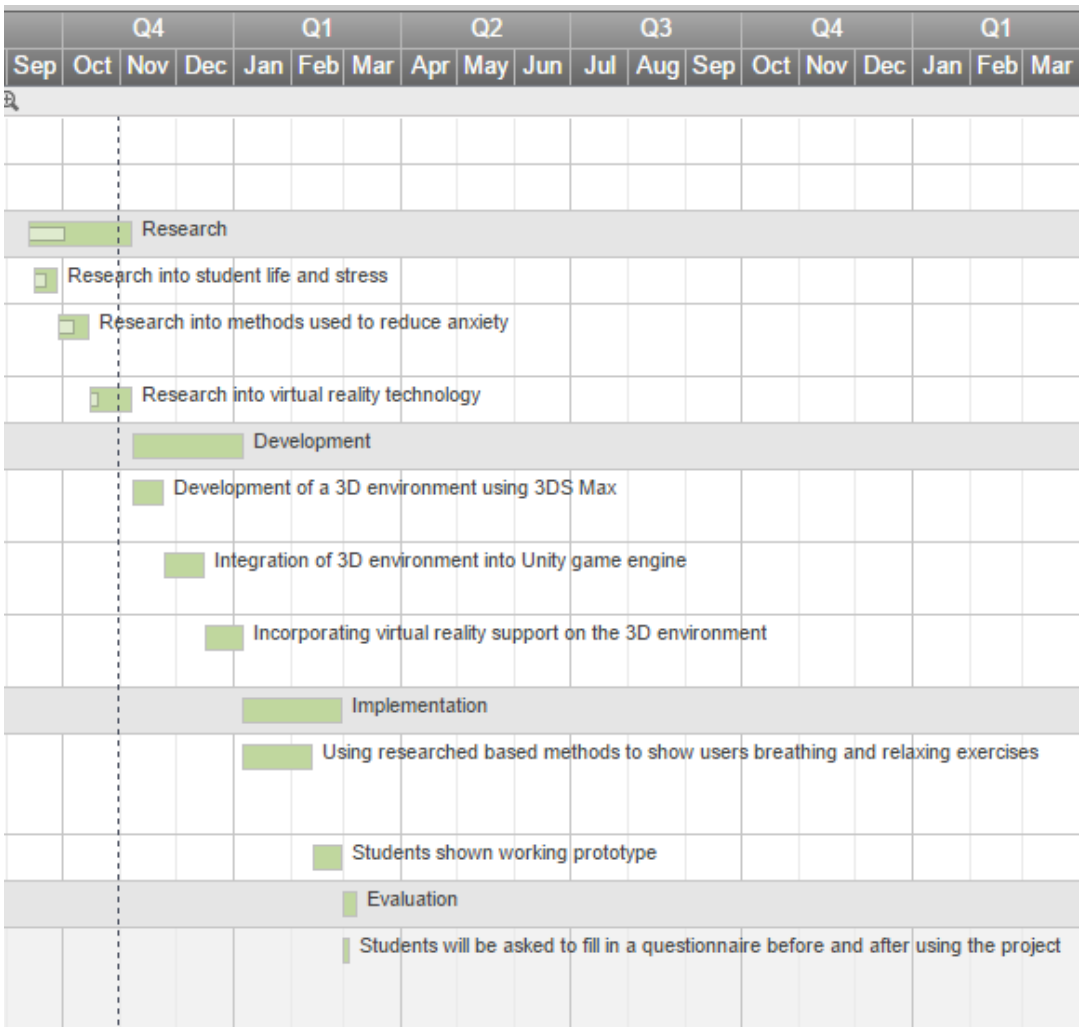
4 Legal, Social, Ethical and Professional

This issue is sensitive so all appropriate precautions will be taken when having users try out the prototype. This project will hopefully shine some light on how virtual reality can be used in a home setting. No one with actual mental health issues should use the project and expect it to cure them; this project will simply be a way to increase the engagement of guided meditation.

5 Planning

The project prototype will be modelled around an iterative prototype design process. This will ensure that the idea behind the project will be followed throughout development including the 3D environment and breathing exercises. A story board will be drawn up showing the menu system and the exercises that the user will see in virtual reality. The 3D environment will be sketched out before being turned into a 3D model.

The project will follow strict adherence to a time schedule which is visualised in the following Gantt chart:


Q4
Q1
Q2
Q3
Q4
Q1
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Research
Research into student life and stress
Research into methods used to reduce anxiety
Research into virtual reality technology
Development
Development of a 3D environment using 3DS Max
Integration of 3D environment into Unity game engine
Incorporating virtual reality support on the 3D environment
Implementation
Using researched based methods to show users breathing and relaxing exercises
Students shown working prototype
Evaluation
Students will be asked to fill in a questionnaire before and after using the project