# Contents

[1.0 Contents 1](#_Toc497125670)

[2.0 Introduction 2](#_Toc497125671)

[2.1 Glossary of Terms 2](#_Toc497125672)

[2.2 Background 2](#_Toc497125673)

[2.2.1 Accessibility in Gaming 2](#_Toc497125674)

[2.2.2 The Problem 2](#_Toc497125675)

[2.2.3 Development Solution 3](#_Toc497125676)

[2.2.4 Justification of Participants 3](#_Toc497125677)

[2.3 Project Outline and Research Question 3](#_Toc497125678)

[2.3.1 Risk Assessment 3](#_Toc497125679)

[2.3.2 Ethics Considerations (as approved by David Moffat) 4](#_Toc497125680)

[2.3.3 Project Development Lifespan 4](#_Toc497125681)

[2.3.4 Research Question 5](#_Toc497125682)

[3.0 Literature and Technology Review 5](#_Toc497125683)

[3.1 Development process 5](#_Toc497125684)

[3.1.1 Waterfall 5](#_Toc497125685)

[3.1.2 Agile 5](#_Toc497125686)

[3.2 Development engine 6](#_Toc497125687)

[3.3 Development platform 6](#_Toc497125688)

[3.4 Blindness in gaming 6](#_Toc497125689)

[4.0 Methods 6](#_Toc497125690)

[4.1 Dev methods 6](#_Toc497125691)

[4.2 Experiment methods 6](#_Toc497125692)

[4.3 Evaluation Methods 6](#_Toc497125693)

[5.0 Remaining tasks 7](#_Toc497125694)

[5.1 Finish Development and debugging 7](#_Toc497125695)

[5.2 Experiment carry out 7](#_Toc497125696)

[5.3 Final report 7](#_Toc497125697)

[5.4 Presentation 7](#_Toc497125698)

[6.0 Appendices 7](#_Toc497125699)

# Introduction

## 2.1 Glossary of Terms

Sight Loss – An individual is described as suffering from sight loss when their visual acuity is found to be 6/12 or worse as per the State of the nation year in review carried out by specsavers in coordination with the Royal National Institute for Blind People (Specsavers, 2017)

Project Sprint – The term used to mean a small but intensive period of work done to a short deadline.

Bugs – An error or flaw in the software system which causes the program to behave in unexpected ways or provide an incorrect result.

## 2.2 Background

### 2.2.1 Accessibility in Gaming

The Video Game industry has been a growing economic powerhouse since its creation. From its simple origins in arcades where each play cost 25 cents to industry worth 99.6 billion dollars in 2016 according to the Newzoo 2016 global games market report (Newzoo, 2016). Despite what is suspected to be continued financial growth, there has been very limited development in the untapped market of accessibility in gaming. With approximately 360,000 people in the UK being registered as blind or partially sighted (1 in 30) as of 2014 according to the Royal National Institute of Blind People (RNIB). The number of those with sight loss are predicted to increase past 2,250,000 as of 2020 as stated by the RNIB for reasons such as an aging population (Transversal, 2014). This increasing population is however comparatively underserved in the current games market with Audiogames.net showing an available 606 audio games at time of writing (Audiogames.net, n.d.).

While there are companies such as Audiogames.net and AppleVis both of which provide gaming accessibility help to those with limited to no sight capabilities. AppleVis provides reviews of iOS games based upon how accessible they are. Audiogames.net has compiled a list of games available and accessible to those with visual impairments, primarily in the field of the audio games genre. Companies like these are comparatively few and far between whereas companies that do not put accessibility at the forefront of their development model are far greater in number.

Development of audio games as a genre is not inherently more difficult than the development of any other genre of games. It does however, require a paradigm shift. Which would put the audio development of a game to the forefront and allow visuals to take somewhat of a backseat by comparison, which up to this point has not occurred.

### 2.2.2 The Problem

It is this is inherent lack of accessibility that is the problem within the games industry. This not only lacks the inclusivity which the industry is often known for but also leaves a large market of potential players untapped. This market, treated by many developers as empty or even non-existent considering that the majority of those suffering from blindness are 50 years of age or older (World Health Organisation, 2010) but the majority of those who play video games are under the age of 36 (Grubb, 2014).

This lack of inclusivity is largely a problem for any who wish to become part of the multimillion dollar competitive gaming industry. This is exemplified by the fact that currently in the industry there are no competitive games aimed at non-sighted individuals. As such, solving this problem and the methods to do so will be the primary focus of the technology and literature review sections of this report.

### 2.2.3 Development Solution

The proposed solution to this problem is the design and development of a game which would allow a non-sighted person to play at equal level with a sighted opponent. The primary solution is to use common audio game development techniques to develop this game and to design the game with accessibility at its core. To do this, research has been conducted on a variety of topics such as; audio game development techniques, possible game development technologies (namely engines and platforms), development processes and the common uses of gaming technologies in the treatment and lifestyles of those who suffer from a lack of sight.

### 2.2.4 Justification of Participants

This research and development project will primarily focus on those with complete blindness under the assumption that if those with no light perception are able to comfortably play the game then those with limited light perception will also be capable of this. This also serves to reduce the variety of required test subjects as well making test subjects easier to find and simplify the testing process. If possible, second stage tests will be conducted to include those with more limited degrees of visual impairment. This will serve to be able to widen the user base by confirming that those with less severe visual impairment are able to play the game. If necessary, though undesirable, test subjects will consist of sighted individuals who are made to wear blindfolds. This is simply a method of managing the worst-case scenario in the instance that no subjects who suffer from no light perception can be found.

## 2.3 Project Outline and Research Question

### 2.3.1 Risk Assessment

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Risk ID | Risk | Risk Severity | Risk Probability | Contingency Plan | Tools Used | Plan to Subvert Risk |
| 01 | Not enough participants to obtain meaningful data | High | Medium | Offer possible participants gifts such as pizza, fruit, juice etc. in order to encourage participation other students. | Participants, lecturers, email addresses | Contact lecturers and request that they send out emails requesting participants and invite students to participate |
| 02 | Game Not Developed Properly (excessively buggy/broken) | High | Low | Run consistent playtesting sessions prior to the experimental sessions in order to ensure bugs are quickly found and fixed. | Game Engine, Computer, Integrated development environment (IDE) | Risk cannot be fully subverted. Minimised through the full knowledge of the code and research of the techniques to be used coupled with time management. |
| 03 | No meaningful conclusions can be drawn from data gathered | High | Low | No contingency plan necessary as this would simply show invalid hypothesis | Computer with SPSS software installed | Successfully analyse data with SPSS |
| 04 | Participants drop out of experiment after agreeing to do so. | High | Low | Offer possible participants gifts such as pizza, fruit, juice etc. in order to encourage participation from other students. | Participants, lecturers, email addresses | Regularly contract prospective participants with reminders about dates and times scheduled |
| 05 | Game incomplete or unfinished | High | Low | None available as if the game is incompletely testing and further experimentation cannot be done | Game Engine, Computer, Integrated development environment (IDE) | Carefully plan stages and follow development timescale throughout the creation of the game. |

### 2.3.2 Ethics Considerations (as approved by David Moffat)

Participants who are over the age of 18 are required to take part in this project. Due to the fact that this development and subsequent experimentation will require human participation, an ethics approval form will be required to be filled out, signed by the project supervisor and subsequently sent off to the ethical approval committee. Receiving ethical approval as early as possible will be essential for the ongoing success of this project.

Autonomy – Any participants who wish to leave at any time during the experiment are welcome to do so. All participants will be advised exactly what this research will be used for and what is required from them at time of request for participation. No one will be persuaded or coerced by researchers or outside influences during any meetings. All participants will be notified that they have the option to leave at any time. All who take part will be anonymous and be advised as such.

Confidentiality – All information relating to participants will be kept anonymous and will not be released. Information will exclusively be used for the project and will be destroyed at completion of project.

Special Needs – All special needs will be taken into account for any participants who require any additional support, such as those who have visual impairment requiring a researcher to guide them to the project area. Participants will be made aware that they have the option for this additional support.

Protected Vulnerable Groups – To ensure there is no conflict regarding protected vulnerable groups disclosure no one who is in receipt of any of the following will be asked to take part in this test; Registered Care Services, Community Care Services, Health and Welfare Services. To determine this, participants will be asked in advance and any who receive these services or refuse to answer will not be considered eligible for the test.

### 2.3.3 Project Development Lifespan

The initial gannt chart detailing the projected development of this project as intended at the time of the project proposal can be found in Appendix A. The new gannt char including the addition of realistic timescales as the project currently stands is found in Appendix B. As shown, the development of the project is being given the most time in order to ensure that the programme itself is completed in time, this stage will also incorporate the testing phase which will stretch into the second semester. After completion of testing to a satisfactory standard, the actual experimental phase of the project will take place. During the experimentation phase participants will be recruited to play the game. Each play session will be timed, and scores recorded using analytics software. These shall all be evaluated to determine the validity of the hypothesis presented in the evaluation stage

### 2.3.4 Research Question

“Can a competitive video game be developed which gives no advantage to users who are fully sighted compared to those whose sight is impaired?”

# 3.0 Literature and Technology Review

## 3.1 Development process

For the development of this project traditional and modern development processes must be considered. For the purposes of this report the traditional waterfall model will be compared with the more model agile methodology.

### 3.1.1 Waterfall

The waterfall development model is a predictive sequential development lifecycle which is split into specific phases. These phases are, in order; Requirement analysis, System design, Implementation, Testing, Deployment, Maintenance. Within this system each phase is only begun upon completion of the previous phase.

The sequential nature of this model means that it is simple to understand and implement, being task oriented in a humanistic view. This model is most useful when the product requirements are explicitly defined and rigidly implemented, this means there are no vague or nonspecific requirements as they are ironed out before any sort of design or development takes place and allows projects to have clear milestones. In this model when less specific details are provided certain phases take longer and become more difficult to implement causing a backlog of the whole process however. This staged approach leads to a lack of any working software until halfway through the life cycle of the project (TutorialsPoint, 2017) which, in the case of a task backlog can put a project far behind schedule. The knock-on effect this can have to the project impedes the testing and evaluation phases in the case that the requirements are not fully specified. This can often be the case in a game as they regularly need to be altered based upon the data retrieved from the evaluation stage of alpha and beta tests. To that effect regularly games employ this model with additional testing phases (alpha and beta) to allow changes to take place after initial development and returning to previous stages when it becomes apparently necessary.

### 3.1.2 Agile

Agile is an adaptive development methodology which takes an iterative cyclical approach to software development.

Each iteration involves aspects of all members of the development team going together through the following stages in order; planning, requirements analysis, designing, building, testing. Each of these iterations increment upon the features of the previous with the final iteration (ideally the release build) holding all required features (TutorialsPoint, 2017). Regular project sprints are done to determine progress of the project and identify any issues as they arise. This is a common methodology for games as it emphasizes regular progress meetings to allow regular bug detection. This can mean that development time is extended however with the testing phase in the cyclical nature of these iterations taking longer than initially hoped. Often considered a good development strategy for video games, this outputs a semi functional product very quickly to allow as proof of concept for client review or pitch, creating a vertical or horizontal slice after each iteration. This model takes an adaptive approach to software development compared to more traditional methods (such as the previously mentioned waterfall method). This development method can unfortunately fall down when complex features that cannot be broken down are introduced. In the case of a feature such as this the development time slows as a bottleneck occurs in whichever team is primarily responsible for that feature.

Despite this, the Agile development methodology is the best suited to the game development project proposed. Applying a version of Agile development methodology tailored for a single person team with self-imposed sprints and limitations for bug-fixing will hopefully prove to be an effective development practice.

## 3.2 Development engine

The engine for the development of this project must be carefully considered. Its adaptability portability must be considered prior to development to allow for maximum outreach. Language and peripheral support must be taken into account to ensure development continues with ease without the necessity of attempting to develop a new programming language or learn an entirely new syntax. Experience with the engine and knowledge of its inner workings should be considered to cut down on development time and allow a greater period for experimentation and evaluation of results. The available features of the chosen engine must be considered for the development of the project and, in the instance that a feature is missing, development time and importance level for that feature must be considered. Furthermore, accessibility of documentation is needed for error handling, should any occur. To this effect the popular development engines Unity 3D and Unreal Engine 4.

### 3.2.1 Unity 3D

Unity 3D is a development engine designed on portability, accessibility and ease of use. It supports the languages C# and JavaScript with the ability to write plugins to be read by the engine in C++. The core of unity itself is written in C++ with certain aspects of the editor being written in C#. This allows the engine to be very portable and gives it a shorter compile time.

Of the supported languages C# allows function overloading, JavaScript does not however which would reduce the customisability of the software to a small degree, however the requirement of function overloading is not a complete necessity as instead of using the process of function overloading a simple case of multiple function names can be used (Rongala, 2017). Unity 3D also a particularly portable development environment allowing porting to; virtual reality, mobile (android and iOS), console, PC, Mac, Linux, tvOS, webGL, Samsung TV and PS Vita. This portability can provide a great deal of outreach to a variety of commercial devices, allowing further development into a commercial marketplace in the instance this project proves successful.

Unity 3D offers a standard set of classes for the development of games. These classes allow a simple and easy starting point for the development of game software. This development environment also provides a series of tutorials for all technology used and there is a large community of user made tutorials available online which allow the creation of a variety of games with minimal effort. These tutorials can act as a structure for the creation of software. However, this resource must not be overused to avoid any issues with plagiarism in the development of this game in an academic context.

Unity 3D provides 2 integrated physics engines, one for 2D physics and another for 3D physics. They both make use of rigidbody components (labelled as “Rigidbody 2D” in the case of 2D physics). The primary difference between the two physics engines is largely the addition of the extra dimension which allows the use of the Z axis (Technologies, 2017). This provides more design opportunities for the development of the project with opportunity to have a 3D or 2D game. This opportunity must also be carefully considered for the development of the project in the instance Unity 3D is the chosen development engine.

### 3.2.2 Unreal Engine 4

Unreal Engine 4 is the primary game engine for the game development and publishing company, Epic Games. Known for developing games such as; Unreal Tournament, the Gears of War series, Bulletstorm and Paragon. All developed in the unreal engine, these games show intuitive use of physics, lighting and mechanical techniques that the unreal engine provides. The engine itself is written in C++ as with Unity3D allowing it to also have a high degree of portability and reduced compile time.

Unreal supports C++ as a development language for the basis of blueprint classes. The blueprint system is a visual scripting system to allow classes to be created in an innovative manner. The blueprints system uses visual blocks of code to show functions and references to variables (Games, 2017). This development method makes it easier for those in non-programming teams to create software for specific features within the game. This however, is not necessarily a benefit in this project as this is development project is a programmer specific project. Meaning the programming itself requires less simplification than in the case where designers would be required for the creation of features.

## 3.3 Development Platform

Oculus touch

Playstation VR

Google Daydream

Computer

Mobile

## 3.4 Blindness in gaming

Blindness in casual Gaming

Blindness in competitive gaming

Using VR for blindness

# 4.0 Methods

## 4.1 Dev methods

## 4.2 Experiment methods

## 4.3 Evaluation Methods

# 5.0 Remaining tasks

## 5.1 Finish Development and debugging

## 5.2 Experiment carry out

## 5.3 Final report

## 5.4 Presentation

# 6.0 Appendices

Appendix A ****

*Figure 1: Gantt chart Generated by Creately Gantt Chart Software (2017)*

References

Newzoo. (2016). *The Global Games Market 2016 | Per Region & Segment | Newzoo*. [online] Available at: https://newzoo.com/insights/articles/global-games-market-reaches-99-6-billion-2016-mobile-generating-37/ [Accessed 13 Oct. 2017].

Transversal, A. (2014). *How many people in the UK have sight loss? - RNIB - supporting blind and partially sighted people*. [online] Help.rnib.org.uk. Available at: https://help.rnib.org.uk/help/newly-diagnosed-registration/registering-sight-loss/statistics [Accessed 15 Oct. 2017].

Specsavers (2017). *The State of the Nation Eye Health 2017: A Year in Review*. [online] Royal National Institute of Blind People. Available at: https://www.specsavers-spectrum.com/wp-content/uploads/2017/09/RNIB-Specsavers-State-of-the-Nation-Report-2017.pdf [Accessed 15 Oct. 2017].

Audiogames.net. (n.d.). *AudioGames, your resource for audiogames, games for the blind, games for the visually impaired!*. [online] Available at: https://www.audiogames.net/list-games/listgames.php?mode=full [Accessed 15 Oct. 2017].

World Health Organisation (2010). *Global Data on Visual Impairments 2010*. [online] World Health Organisation, p.Page 5. Available at: http://www.who.int/blindness/GLOBALDATAFINALforweb.pdf [Accessed 15 Oct. 2017].

Grubb, J. (2014). *Gaming advocacy group: The average gamer is 31, and most play on a console*. [online] VentureBeat. Available at: https://venturebeat.com/2014/04/29/gaming-advocacy-group-the-average-gamer-is-31-and-most-play-on-a-console/ [Accessed 15 Oct. 2017].

TutorialsPoint (2017). *SDLC Waterfall Model*. [online] www.tutorialspoint.com. Available at: https://www.tutorialspoint.com/sdlc/sdlc\_waterfall\_model.htm [Accessed 25 Oct. 2017].

TutorialsPoint (2017). *SDLC Agile Model*. [online] www.tutorialspoint.com. Available at: https://www.tutorialspoint.com/sdlc/sdlc\_agile\_model.htm [Accessed 27 Oct. 2017].

Rongala, A. (2017). *Benefits of C / C++ over Other Programming Languages*. [online] Invensis Blog. Available at: https://www.invensis.net/blog/it/benefits-of-c-c-plus-plus-over-other-programming-languages/ [Accessed 31 Oct. 2017].

Technologies, U. (2017). *Unity - Manual: Physics*. [online] Docs.unity3d.com. Available at: https://docs.unity3d.com/Manual/PhysicsSection.html [Accessed 31 Oct. 2017].

Games, E. (2017). *Programming Guide*. [online] Docs.unrealengine.com. Available at: https://docs.unrealengine.com/latest/INT/Programming/index.html [Accessed 31 Oct. 2017].