Plymouth University

School of Computing, Electronics and Mathematics

PRCO304
Final Stage Computing Project
2017/2018

BSc (Hons) Computing and Games Development

Ludwik Bacmaga

10595124

UX in AR Games

Supervisor: Mr Marius Varga
Second Marker: Dr Nigel Barlow

Acknowledgements

To be completed…

Abstract

To be completed…

Table of Contents

Word Count: ##,###

1. Introduction

Project focuses on creating developing an Augmented Reality, AR, game with well implemented User Experience, UX. This is achieved by using ARKit technology, created by Apple, and using ARKit Unity Plugin to implement it using Unity 3D game engine. Purpose for this project was to research and develop good UX for AR game, since it is still a new technology.

UX consists of different features, namely: User Interface, Usability, Application Interaction, and any other features that influence user satisfaction with the game, application. By changing implementation of a feature, it will affect the UX; this project aims in finding UX that is enjoyable for the user in AR game.

Creating good UX will be created by creating prototypes, followed by testing and gathering feedback on how to improve on the prototype. The core mechanic of the game would be kept the same, just the implementation would change based on the feedback gathered. By the end, from feedback gathered and also from experience by playing the game, it should be presentable if UX does impact enjoyment for the player.

2. Background, Objectives & Deliverables

2.1. Background

User Experience, UX, is very crucial for a game, or application, success. Especially within the new technology which is still evolving, like Augmented Reality, AR. Augmented Reality, AR, combines virtual with the real world using devices such as mobile phones.

Before ARCore and ARKit come out, all other kits that offer development in Augmented Reality required a marker to operate, however, ARKit and ARCore focus on markerless implementation. With this new implementation, there are different ways to implement AR and it is important to utilise UX to take advantage of this new technology.

UX takes into account different things, but it focuses on end user experience. It is accomplished by the way that information is presented to the user by the application:

- User Interface, UI
- Usability
- Application Interaction
- User Research and Data
- User Testing

UX uses all of these features to do research and create prototypes, for final product to be user-friendly and intuitive for people to want to come back and use the application again.

2.2. Aim and Objectives

Aim for this project is to create a game in Augmented Reality, with a good user experience. It is accomplished by testing out different prototypes and researching existing titles.

Objectives for this project were:

- Gain experience using Unity and ARKit package within Unity
- Experience with Xcode and application development for iOS platform
- Better understanding of UX in AR
- Scope out project milestones for project development
- Do prototypes and testing along the development stage

2.3. Deliverables

The deliverables for this project is an iOS game application, implemented in ARKit, which focuses on user experience, UX. Details can be found in section 7.

3. Research

To be completed…

(Convert from mindmaps to text)
Sections:
UX definition, found uses and research from papers, how in theory should be implemented for best use.
UX in current titles – comparable table… what features were tested in this project, what work what doesn't.

For this project, *Standards for Augmented Reality: a User Experience perspective (Reference #)*, and *Visualising Graphs in Three Dimensions (Reference #)* research papers were read and studied which affected how the project was implemented. *Visualising Graphs in Three Dimensions* research paper focuses on how information should be presented to be more distinctive for the human eye. The perception of information. *Standards for Augmented Reality: a User Experience perspective* research paper focuses on how user interacts with AR technology directly and how the information is feedback to the user. Those papers were beneficial in developing this project further and to have successful outcome. Moreover, existing game titles were looked at and evaluated to figure out what work well and what doesn't in those titles, adapting positive, relevant, features to this project.

3.1. Visualising Graphs in Three Dimensions

To be completed…
(USE GRAPHS FROM THE RESEARCH THAT IS USED IN THE PAPER)

Visualising Graphs in Three Dimensions, focuses on graph representation to the user to maximise number of nodes that are distinguishable for the human eye. It compares 2D representation to 3D representation which illustrates the difference and amount of detail that 3D offers compared to 2D representation.

3.2. Standards for Augmented Reality: a User Experience perspective

To be completed…

Standards for Augmented Reality: a User Experience perspective research paper evaluates different ways to interact with AR application, and what outputs are there for application to give feedback to the user. It depends on the application, and what it aims to accomplish, that decides which input and output are used to interact with the user.

3.3. Existing Games

To be completed…
(TABLE THAT WAS CREATED IN COMPARING EXISTING TITLES)

3.4. Summary

To be completed…

In summary…

4. Method of Approach

4.1. Approach

The production of the prototypes and tests will follow an Agile Methodology.

By the use of User Story Mapping, the sprints will be created to set deadlines and have a clear direction what is being implemented at each stage. Each sprint will have specified information with functionality and how it affects the game.

After the sprints are specified, the prototype will be created, followed by quality assurance. Quality assurance will test out the prototype to see if it is suitable for use. Proof of concept will be performed afterwards, which will involve demonstration of the prototype to interested parties and project supervisor to get feedback and make sure prototype is performing as it should.

This method of approach will be repeated, once the technology is successful in testing. The process will be repeated until all agreed features are created. Followed by testing from colleagues and have public testing, where the objective being to get as much feedback as possible.

Technologies that were used for this project are specified in section 4.2.

4.2. Technologies

Unity 2017.3 – Game Engine
Visual Studio 2017 – Script Editing Software
Adobe Photoshop CC – Graphical Editing Software
macOS High Sierra, Version 10.13.4 – OS of machine
Xcode 9.3 – Install program for iOS
Unity ARKit Plugin, version 1.0.14 – Package used to implement ARKit in Unity
iOS 11, iPhone 7 – Device it was tested and implemented on.

4.3. Project Management

Trello is the project management online tool that is used to keep track of the project – that is where the sprints are detailed.
GitHub is the version control tool used for this project.
Weekly highlights are created to keep track of progress of the first 7 weeks of implementation. Daily developer diary contains daily entries with details of what has occurred on that implementation day.

5. Legal, Social, Ethical and Professional Issues

Copyright:
Assets
SFX
Logos

Ethical:
Public testing – approval form

Professional:
Space issue

COPYRIGHT

- *State copyright segment here*

Assets, sounds, and anything else that is used in the project is projected by the copyright. Before using any resources created by others, the copyright was checked, with the author, to make sure that it is legal to be used for this project.

- Have to check what copyright states for Logos, assets and sounds.

Since the Assets, Sounds, Logos are not used for commercial use, no copyright laws were broken in developing this project.

- *State any ethical law (if there are any)*

In this project, there were testing completed which involved other people. This is covered in University of Plymouth Ethical Form. Any participants that took part in testing, and provided feedback, signed the Ethical form to give consent to use their feedback in this project research.

- *State any space relating laws and personal space*

With the use of Augmented Reality, AR, and with this game design, it might be necessary to have spacious environment to be able to use and test the game. It can cause issues when playing this game in public, since it might cross with other individuals personal space. Therefore, it is recommended to play this game where plenty of free space is available. In the development of this project, no personal space was invaded, and at all times the project was tested and developed on, there was plenty of empty space available which did not impact other people movement or space.

6. Implementation

6.1. Process

Initially, organisation was completed on Trello, to keep sprints organised and to have a clear idea what was being accomplished on each sprint. That is also a place where all feedback from Highlights, and any comments to take in to consideration were dropped down, followed by a change in sprint, or future plans, if necessary.

The idea was created by looking at already existing titles, in AR, mobile and computer games. Based on research, the game idea was created, which was split into sprints. The idea evolved with each sprint and feedback, to create more relevant game, and at the same time created a more enjoyable experience for the user.

Following the sprints, it took five sprints to create a playable prototype, which was followed by a testing session with colleagues. At that session, the feedback was gathered, and after analysis the feedback, the UX was not that well implemented, as the game required the user to do extra steps to complete a task, such as to collect ingredient or to activate ability. With that feedback taken into the account, the next sprint was altered to change the game, regarding UI and how the information is presented to the player, with feedback being more positive after the sprint.

Final implementation shows more features of what the current titles in AR are representing, with the more enjoyable user experience by the way that the information is presented and shown to the player. UI is clean, to have as much viewing screen as possible; the player has an option to view instructions how to play the game by clicking help button, this doesn't enforce tutorial on the player if they don't want it; sound is used as atmosphere representation, representing the healthy and infected planet, as well as when infection takes place.

Each sprint was implemented with iteration of different implementations and trying different way of implementing a feature, for example the spread of infection, or the abilities functionality. The implementation that worked the best, and was optimal that work with the rest of the system was kept, and once this was implemented the next feature was being worked on. This project didn't allow for much time to work on graphical implementation of the game, therefore the assets from Unity Assets Store, and online assets, were used to implement the game. All copyrights were checked to make sure it was within copyright to use the assets for this project. Adobe Photoshop was used to create any graphics that I was required to make.

6.2. Requirements

Minimum requirements for this project are:

- iOS11
- Xcode 9
- Apple A9 processor (iPhone 6S/SE and newer, iPad 5th Generation and newer)
- macOS High Sierra

6.3. Design

Design of the game evolved with each implementation of the game. The core mechanic of the game, infection spreading across the planets with player goal is to stop it, has stayed the same, however, the way that it is implemented has evolved.

Initially the game was implemented with retro style, where instructions were only present at the beginning of the game and no way of viewing them again once the game has started. Also, the game was filled with menus. Menus were present to do any action in the game, when the player clicked on the planet, or the pause screen (which was also an inventory screen). This did not feel like a pleasant implementation, and also from feedback the players didn't feel like they know what is going on and did not have a good user experience.

This design has evolved to contain minimum number of menus, while still presenting all the information to the player in different manner. The way that the game instructions are presented to the player has also changed. The player starts to play the game normally, with a help button on the side which shows the instructions of how the game operates. This allows the player to reference back to the help whenever they need to, while at the same time the help, tutorial, screen is not forced on the player when they start the game.

The tutorial screen is available to play through, which is optional and it is implemented with timed events which introduce the player to the game mechanics slowly.

The UI contains help button and the pause button, which is clean and maximises the amount of screen available for the gameplay. The game itself shows abilities using particles and prefabs rotating around the planet, which was previously implemented as a text information in a menu. The colours of particles change depending what ability is available to take. Also, the initial platform has more use, as it will hold the available ability that the player can click on to activate.

Planets have different texture which indicate the state that the planet is at, where it is colourful at the healthy state, contain orange shade when it is at risk of being infected, and red shade when it is infected. This creates a visual representation to the player, which allows player to know what is going on in the game without having any input. Previously most information was presented in the menu system once the planet was clicked on. Moreover, to improve the feedback that game has with the user, once the player has click the planet, the planet now deforms which gives feedback to the player that the planet was interacted with.

Sound was also implemented to indicate the state of the planet, which player can hear depending on the position that they are at. Sound ques are also created to indicate when planet is infected. This improved the playability and user experience, which is also visible on the feedback, compared to the initial implementations where the sound was non-existent.

Assets that are used in the game also improved, as with the time limitation and how much time it takes to implement the game, the assets from the Assets Store were used to improve the graphical aspect of the game. This has improved the visuals greatly.

6.4. Prototype

Prototypes were created to test out the implementation, mechanic, and features in the game. Each sprint had a prototype created, and after sprint five the prototypes have created a playable game experience. Upon that prototype, the game was improved. Each sprint followed this formula: Minimum Viable Product (MVP), Testing, Results from the testing.

6.4.1. Minimum Viable Product

Minimum Viable Product was the minimum implementation of the sprint before completing the sprint. In this project, the MVP was used to create a basic form of the mechanic, the game, but it didn't necessarily contribute in creating an enjoyable gameplay experience and a good user experience. Initial implementation was focused more on implementing the mechanic of the game, however as the project is focused around the user experience, later in the project the focus changed to implementing more enjoyable user experience rather than the mechanic, although initial implementations did allow to test the mechanics of the game.

The MVP for sprint one, sprint specified in section 7.1, was to have a game design that the project will be based on, and at the same time test out the technology, ARKit with Unity. The game design was an idea that was developed further, and changed, throughout the project. Core of game idea stay the same which was to have islands and infection spreads through them, corrupting the islands and the player has to stop it. The initial idea did consist of additional character in the game that the player controls. The technology testing was a tracking test, to see how well ARKit tracks and what could be potentially implemented with this technology. Alongside this, the organisation of time, deadlines, and rough plan of the project was created.

MVP for sprint two, sprint specified in section 7.2, was to create a prototype of islands spawning, and user interaction with the game. The character was implemented in the game, which provided feedback to the user if the character is in correct position (planet), based on where the user has clicked. Basic meshes were used for the implementation and textures to save time. This MVP tested out the islands spread and tracking and player input movement between the planets.

Sprint three MVP, sprint specified in section 7.3, was to implement the infection, the virus, that would spread towards other planets. With an indicator of infected planets. This was implemented using different textures, representing infected planets in red colour, and having a solid line link between the planets, also in red, showing the spread directions. The basic UI, pause screen, was implemented. Start of ingredients spawning on planets, and inventory system was to be started in this sprint too.

Sprint four MVP, sprint specified in section 7.4, focused on completing abilities, with inventory system, to be ready for testing and to gather feedback. This involved ingredient system working, with the solutions created from the ingredients will be placed on the planets to fight the infection. Inventory system was implemented in the pause screen, to save screen space.

Sprint five MVP, sprint specified in section 7.5, was focused on testing the application between colleagues and gather feedback on the game. Analysis of feedback to be completed, and based on feedback alter the following sprints to include the feedback gathered.

Sprint six MVP, sprint specified in section 7.6, focused on altering the game in any way from the feedback gathered that would create a more enjoyable game. This involves improving graphics, UI, and gameplay, even mechanics were to be changed if it required. This sprint focused on creating best possible game with the time provided which focuses on user experience and delivering positive user experience.

6.4.2. Testing

Testing was done after each sprint, and whenever new mechanic or feature was implemented. It was tested on a development device, iPhone 7 (mentioned in section 4.2.), fixing any bugs that were encountered. Major testing was done after sprint 5, which tested the whole prototype with other people gathering feedback while at the same time good moment to monitor any bugs that were encountered to be fixed later.

Apart from bug testing, another reason for testing was to test if the feature implemented works well with the game. This was particularly tested while public testing took place, where feedback gathered reflected the features implemented, and how it could be improved or changed.

6.4.3. Results

Results from testing were evaluated and based on the results the actions took place. Testing took place to get results for the bugs that were in the game but also on the features implemented and how well do they operate in the game. The feedback gathered where the results from the testing and the results showed what features worked well and which should be modified to improve user experience.

After sprint testing, the bugs that were found were fixed, and they varied, which can be seen in developer diary (Appendix I). Public testing indicated results for the features and gameplay, which showed that the UI had to be changed to be more logical and didn’t require player to do unnecessary steps to complete an action. The ingredient system wasn’t that enjoyable, which was changed towards abilities, and how the information is represented in the game was to be changed to be more attractive to the user. Also, implement the sound to enhance user experience. Based on those feedback, the game was changed.

6.5. Control Plan

Within the user experience, the features that were controlled were: User Interface, Game Interaction, and Visuals. All of which impact the user experience and those were changed to enhance the experience for the user while keeping the core mechanic of the game the same.

6.5.1. User Interface

User Interface, UI, is important in AR, as it impacts how the user interacts with the game, and it can also impact how the information is presented to the player. The UI for this project was influenced by the research of already existing titles of AR and other technologies.

Initially the UI was implemented to have many menus, with screen UI to be clean, only pause button being displayed on the screen at all times. The pause menu would be combined with the inventory system to minimalize how much space UI takes on the screen. The menus for planets would display next to the planet to indicate the information about the planet. However, after testing, this implementation produced poor user experience, with feedback gathered stating that user had to do extra steps to complete basic task, where the steps to complete the task could be completed with one or more less actions.

Upon feedback, the final implementation involved game to be more visual and uses visual assets to show information for the player, with help (tutorial) and pause button living on the screen UI at all times. This implementation gathers more positive feedback delivering better user experience.

6.5.2. Game Interaction

Game Interaction focuses on how the game interacts with the user, the player. This involves the player input, how the game presents data for the player, the feedback provided based on the input.

In this project, game, the game interacts with the player by slowly introducing game features to the player, while still allowing players to view how to play the game by pressing help button. There is also visual feedback to the player, opposed to text or menu, which a player can view to see a state of the game. Also, there is visual feedback in the form of mesh deformation that takes place once the player clicked on the planet. There is also sound implemented, which is a way for a game to indicate to the player when there is a danger while also being used for atmosphere creation.

Player can only interact with the game by the use of a click, and the abilities that are in the game. With all of those game interactions implemented, it creates an enjoyable user experience, based on the feedback.

6.5.3. Visuals

Visuals are final feature that can be controlled to impact the user experience. Initially, basic meshes with basic colours were used in the implementation of the game. As initial plan was to create assets in Blender. However, due to time constrain, assets were gathered from Assets Store. Also, abilities are represented in the game in more appropriate manner, using prefabs and particles which enhance the game appearance. This is also how the game represents some information towards a player, which has more positive feedback unlike the initial representation where it didn’t utilize visual aspect that well. The game uses visuals to show the player the state of the game, abilities available to be gathered, and danger incoming to the planet. With visuals, sound is used, to create more engaging and enhanced user experience.

7. Stages

The following stages are the milestones, sprints, that were created to complete this project. They were used to keep the project organised, and helped to make sure that project will be completed on time.

7.1. Stage 1: Game Design & Technology Analysis

First stage involves project planning and creating a draft of sprints that will be used in development of the project. This was done on Trello, by creating a series of sprints that are milestones for the project. Each sprint took into the account the knowledge of the subject and what had to be completed to make significant progress. The sprints were to be edited when one sprint was completed, to keep the project plans up to date.

That is also the initial stage where the research has started, looking at existing titles that can influence the outcome of this project. Research of literature also started at this time, which carried on throughout the project, taking notes on Trello any relevant literature that is relevant to this project.

First stage was also dedicated on figuring out and creating a game design that will be used for the game to be implemented in this project. This design was creating by looking at existing titles and gather up ideas of what game mechanics would work well with AR.

Game design that was the project based on was on creating a game where the player specifies a spawn location. Once the player decides spawn location, the virtual world (planets) are spawn spread apart of each other. Each planet containing an ingredient of some kind to find evil that will appear in the world. In this game, the evil is the infection that spreads across planets and ingredients are the only way of preventing the spread, and fight the infection. Winning condition being get rid of the infection, and losing condition is where all planets are infected. In this game design, the player is forced to move around, taking advantage of ARKit, while it also allows to test different UX implementation on which this project is focused on.

Upon deciding what game will be implemented, the prototype of ARKit plugin working was implementing, main goal being to test out how the markerless implementation works and how well does it track the virtual world. Previously done projects in Vuforia, where the marker had to be used to play, the ARKit markerless implementation offers better tracking without any additional components. This allows for bigger range gameplay, where the camera doesn’t have to view a marker.

With ARKit, the initial prototype was very simple, having specify and spawn location (world centre), followed by the world spawning around the point. To spawn a location, the horizontal plane has to be scanned, where the code for that is already present in the ARKit Unity Plugin. Upon completion, the tracking was good, where player could view different sides of the world without losing track. Having fundamentals in place, the game prototypes and implementation could begin.

7.2. Stage 2: Game Implementation

Second stage that was set for the project was to begin game implementation. It involves implementing the user input, and the virtual world spawn. It was accomplished with taking initial prototype that was used to test out the technology, which then was improved by spawning multiple planets. With the world spawned, that doesn't collide with each other, the user input testing was implemented. It was done by creating a character that jumps to a planet that the player has clicked on.

The character was later in the testing and implementation deleted, as it is unnecessary for this game design. However, the character implementation did assist in making sure that the game interprets the player input correctly, as there was visual character jump feedback to the planet that the player has clicked on.

This also tested out the functions that were used for the implementation, fixing any bugs that may have appeared along the way. Knowing that the functions work as intended, once the character was removed from the implementation, those functions were reused to interpret player input.

7.3. Stage 3: Game Mechanics

In the Stage Third of the project, the game mechanics were implemented. The main mechanic, the infection spread, was implemented initially with the planets being connected together by a link, in game it is represented with a prefab that is rescaled in the correct directions. This creates a link between the planets, with a visual red colour implementation that symbolises danger. At this point, the character was still implemented, and once the spread reaches the game character, the player loses the game. This game design was a bit changed later after implementing and testing the whole prototype. As the character was completely removed.

Alongside the spread, the winning condition was also implemented, the ingredients, that could be gathered to fight the infection. Originally the idea was to gather the ingredients from planets and once the player has enough ingredients, the player is able to construct one of three abilities. Initially there was a bomb, permanent health link, and temporary planet shield.

Initially, the ingredients system was implemented in UI that takes a lot of screen space, and adds additional stages for the player to do which were at the end unnecessary. Once the player clicks the planet, the menu system would be displayed next to the planet which would show what player is allowed to do on the chosen planet. The pause menu was the place where the ingredients information was stored.

7.4. Stage 4: Complete Game Prototype

Stage Four was dedicated on completing the initial prototype to be ready to be used for public testing. The prototype would involve a minimum viable product – UI, working game mechanics (spread, ingredients, crafts system, win/lose scenario). It did also contain introductory instructions, which described everything in the game. However, it was clear that without describing the game, the instructions were unclear.

The graphics for the prototype were very simplistic, as there was no time for complex asset creation, or any animations. Therefore, it didn't look very inviting and was very basic. There were also few bugs that still existed which were not fixed until the next stage. However, for testing and prototyping, it did display the mechanics and how the game would work, with initial UX design with plenty of menus.

7.5. Stage 5: Testing & Feedback Gathering

Stage Five was testing and gathering feedback. Google Forms were used to gather the necessary feedback, with an option to analyse the results at the end. The goal of having minimum of 10 people testing out the game, also signing the ethical approval form.

In testing, the game was presented in front of the testers, where every person played the game, followed by filling out feedback on google forms. The feedback gathered were similar. Feedback being that all the menus were unnecessary where the player had to make additional clicks, moves, which could have been done quicker, with cleverer implementation. Moreover, the feedback also involved the character is not necessary, as well as the instructions at the beginning were not very clear and the visual representation of information could be easier represented.

As it is project about UX, it is important that the information is presented to the player is enjoyable manner, and the gameplay is logical. It involves the introductory segment at the beginning of the game, tutorial, that present that player to the game mechanics and gets familiar with the game. Next stage focused on improving the game prototype, changing game design accordingly, improving the UX in the game.

7.6. Stage 6: Improvements Implementation

Stage Six was focused on improving on prototype that was created. The improvements that were implemented from the initial prototype are as follow.

- UI is simplified, clean, on the screen only showing the pause button, and an indicator on how many planets are healthy and how many are infected. No other UI is required for this type of game, as from testing and feedback gathered the new implementation uses more of AR technology.
- The character on the planets was removed, as it is unnecessary for this sort of a game. The camera position, the player, is now the only player that plays the game, no other characters to control. This, like with the UI, has good feedback, saying that it uses AR technology more and it is more enjoyable to play.
- There are no more menus next to the planets to indicate the possible actions for the player. The player is able to click on the planets to collect the ability, and use the ability on a specific planet, but apart from that there is no menus. The abilities are indicated by the use of orbiting prefab, with different ability showing different effects. This is more visibly pleasing, also visible to the player without doing any actions. With this implementation, no extra menus are required, and controls are simple and logical.
- There is a mesh deformation applied to the planet when the planet is clicked, for visual feedback, and to enhance user experience.

- Music is added for better engagement with the player and to make the game more atmospheric. Music is used also as an indicator to indicate the distance between objects, as the sounds become quieter based on the distance.
- The planets, and projectiles, are now prefabs taken from the Assets Store, which improve the visuals of the whole game. Overall, particles, improved graphics and animations are used to enhance visual aspect of the game.
- No more introductory instructions at the beginning of the game, it is replaced with timed sequences which present the game to the player. Like a tutorial. It presents different mechanics of the game to the player in the more interactive manner.

All of those improvements did equal to improved feedback, and better user experience in general. That is what this project focuses on.

8. External Libraries

Following external library was used in this project: Unity ARKit Plugin version 1.0.14. This library was used to use ARKit in Unity and build the project, followed by compiling and installing it on iOS developer device.

9. Project Evaluations

9.1. End Project Report

To be completed…

9.2. Project Port Mortem

To be completed…

9.3. Objective Evaluations

To be completed…

10. Conclusions

To be completed once development been finished, or close to finishing…

10.1. Lessons Learnt

TBC…
Learnt:
Time management,
Organisational Skills,
Unity,
ARKit,
UX,
App implementation,
C#,
Software development

10.2. Further Development

TBC…
Further development:
Asset change – more fitting theme, create own,
Different level design,
Progression system,
Rewards for completion level,
Menus,
Polishing it up

10.3. Conclusion

TBC…
Conclusion:
Prototyping to test what work and what doesn't,
Finished with well-designed UX in game in AR,
Combined features that work and removed what didn't,
Achieved what was aimed for,
What works in AR UX and what doesnt

11. References

To be completed…
Fill out references and link them to the paper. (links and sections mentioned in developer diary)

Appendices:

Upload documents for the Appendices…

Appendix A: User Guide

Appendix B: PID

Appendix C: Highlights

Appendix D: Stage Objectives, Plans and Reviews

Appendix E: Exception Reports/Plans

Appendix F: Research Documents

Appendix G: Ethics Application

Appendix H: Ethics Approved Application

Appendix I: Developer Diary

Appendix J: Test Results