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

I’d like to give my thanks to all those who have encouraged and influenced this project, through participating in testing or just providing ideas for improvements.

Special thanks to Marius Varga, as my project supervisor, supporting me any way possible throughout this project, making it as good as possible, pushing me to try my best.

I would also like to thank my girlfriend, who offered emotional support throughout the project and for listening to me even when she didn’t know what I was talking about.

# Abstract

Augmented Reality, AR, is a technology that haven’t been explored much as of yet when it comes to game development, or application development, as it is still, debateable, a new technology. There are still areas which can be improved on, but for application success what is very important is to have very good user experience, UX. This research explores this area and tries to implement an AR game, using ARKit, with a well-designed UX that would be enjoyable for the user.

This report discusses in detail the processes used throughout the project, starting at the background and research of AR and existing titles, to specifying the requirement and the process that it took to complete this project. Every stage of implementation is explained, with the design choices that were made in the implementation stage. Comparing two implementation that were completed for this project, which compares good UX implementation to the bad one.

Report explains each stage of the implementation, where the report is finished with the project evaluation of the work completed. This is done through project evaluation, future improvements, objective evaluation, lessons learnt in the project, project port-mortem, and conclusions gathered at the end. The purpose of those evaluations is to evaluate project performance, as well as to see whether the objectives been met. As well as to evaluate what went well and what could be improved in this project.

# Table of Contents

[Acknowledgements 2](#_Toc514169230)

[Abstract 2](#_Toc514169231)

[Table of Contents 3](#_Toc514169232)

[1. Introduction 5](#_Toc514169233)

[2. Background, Objectives & Deliverables 5](#_Toc514169234)

[2.1. Background 5](#_Toc514169235)

[2.2. Aim and Objectives 5](#_Toc514169236)

[2.3. Deliverables 6](#_Toc514169237)

[3. Research 7](#_Toc514169238)

[3.1. Visualising Graphs in Three Dimensions 7](#_Toc514169239)

[3.2. Standards for Augmented Reality: a User Experience perspective 8](#_Toc514169240)

[3.3. Existing Games 11](#_Toc514169241)

[3.4. Summary 13](#_Toc514169242)

[4. Method of Approach 13](#_Toc514169243)

[4.1. Approach 13](#_Toc514169244)

[4.2. Technologies 14](#_Toc514169245)

[4.3. Project Management 14](#_Toc514169246)

[5. Legal, Social, Ethical and Professional Issues 15](#_Toc514169247)

[6. Implementation 16](#_Toc514169248)

[6.1. Process 16](#_Toc514169249)

[6.2. Requirements 17](#_Toc514169250)

[6.3. Design 18](#_Toc514169251)

[6.4. Prototype 19](#_Toc514169252)

[6.4.1. Minimum Viable Product 19](#_Toc514169253)

[6.4.2. Testing 20](#_Toc514169254)

[6.4.3. Results 20](#_Toc514169255)

[6.5. Control Plan 20](#_Toc514169256)

[6.5.1. User Interface 21](#_Toc514169257)

[6.5.2. Game Interaction 21](#_Toc514169258)

[6.5.3. Visuals 22](#_Toc514169259)

[7. Stages 23](#_Toc514169260)

[7.1. Stage 1: Game Design & Technology Analysis 23](#_Toc514169261)

[7.2. Stage 2: Game Implementation 24](#_Toc514169262)

[7.3. Stage 3: Game Mechanics 24](#_Toc514169263)

[7.4. Stage 4: Complete Game Prototype 24](#_Toc514169264)

[7.5. Stage 5: Testing & Feedback Gathering 25](#_Toc514169265)

[7.6. Stage 6: Improvements Implementation 25](#_Toc514169266)

[8. External Libraries 27](#_Toc514169267)

[9. Project Evaluations 27](#_Toc514169268)

[9.1. End Project Report 27](#_Toc514169269)

[9.2. Project Port Mortem 27](#_Toc514169270)

[9.3. Objective Evaluations 28](#_Toc514169271)

[10. Conclusions 29](#_Toc514169272)

[10.1. Lessons Learnt 29](#_Toc514169273)

[10.2. Further Development 29](#_Toc514169274)

[10.3. Conclusion 29](#_Toc514169275)

[11. References 30](#_Toc514169276)

[Appendices: 31](#_Toc514169277)

[Appendix A: User Guide 31](#_Toc514169278)

[Appendix B: PID 31](#_Toc514169279)

[Appendix C: Highlights 31](#_Toc514169280)

[Appendix D: Stage Objectives, Plans and Reviews 31](#_Toc514169281)

[Appendix E: Exception Reports/Plans 31](#_Toc514169282)

[Appendix F: Research Documents 31](#_Toc514169283)

[Appendix G: Ethics Application 31](#_Toc514169284)

[Appendix H: Ethics Approved Application 31](#_Toc514169285)

[Appendix I: Developer Diary 31](#_Toc514169286)

[Appendix J: Test Results 31](#_Toc514169287)

**Word Count:** ##,###

# 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.

# Background, Objectives & Deliverables

## 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.

## 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

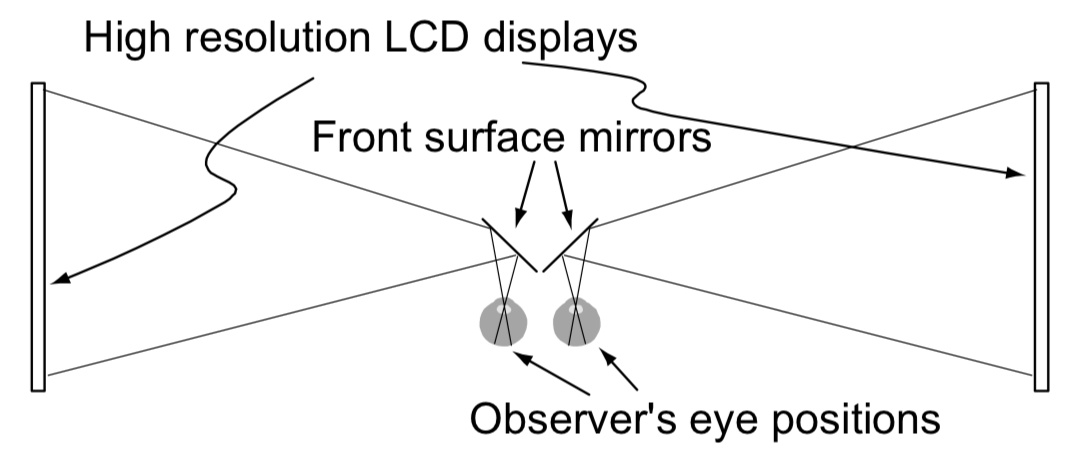
## Deliverables

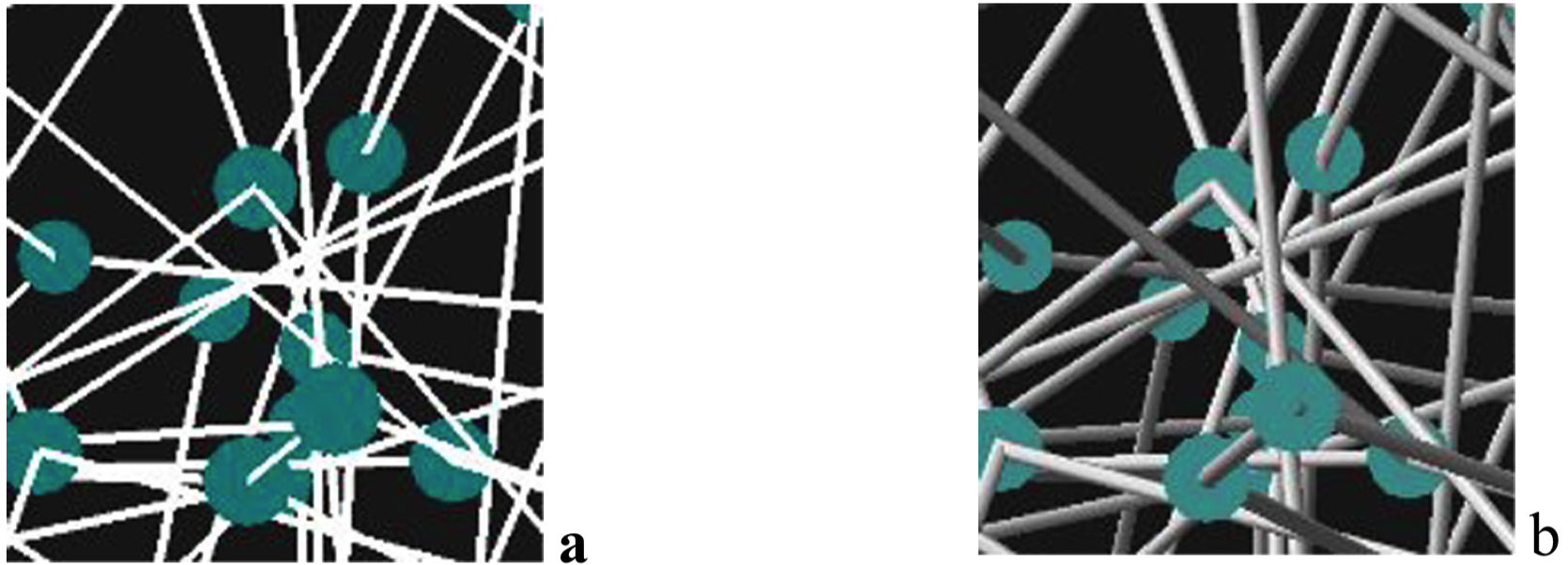
The deliverables for this project is an iOS game application, implemented in ARKit, which focuses on user experience, UX. Details can be found in section 7.

# Research

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.

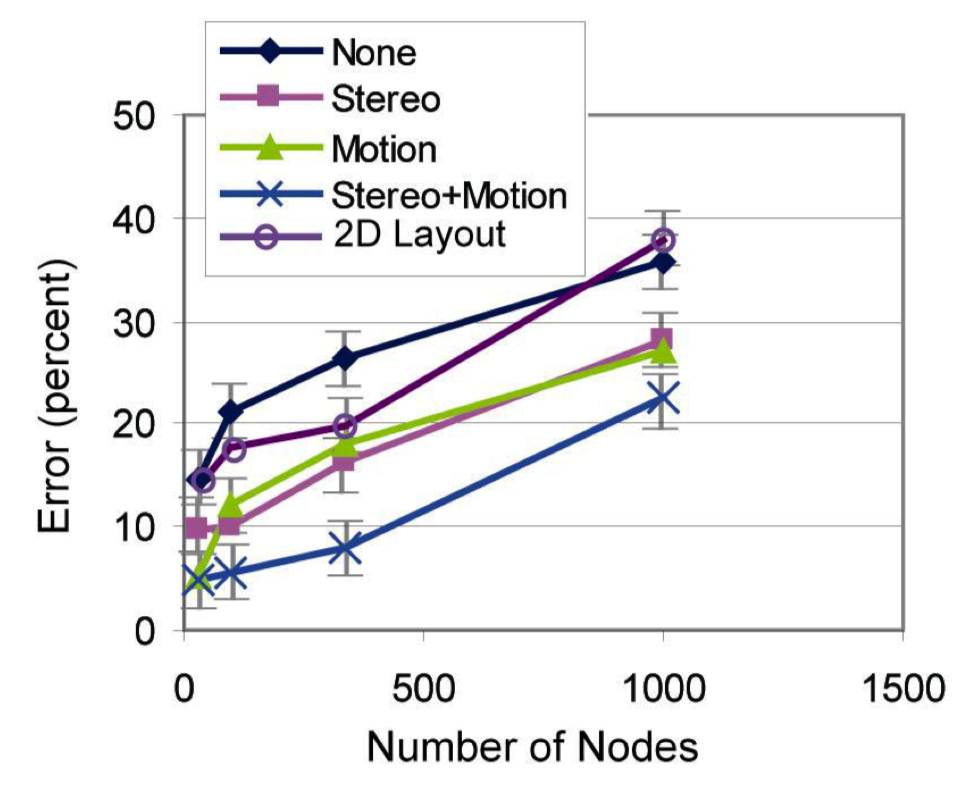
## Visualising Graphs in Three Dimensions

*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.

*Figure 1. A Wheatstone stereoscope arrangement provides a ghost-free display.*

*Figure 2. Snapshot ‘a’ shows 2D graph representation, snapshot ‘b’ shows 3D representation.*

*Figure 1* illustrates how the test was constructed, by a person looking at graphs, as seen in *Figure 2*. The response time, and correct answers were recorded, and based on those results the conclusions were made. Each test was done under different conditions, using different techniques to help tester get correct results. The conditions were: stereo, motion, rotation, 2D and 3D image. *Figure 3* shows the error rate as a function of graph size.



*Figure 3. Errors as a function of graph size. Average data based on 14 participants.*

Based on the results from *figure 3*, having 3D representation s more distinguishable to the human eye to detect. If it is combined with motion and stereo, the error rate is drastically reduced. From that research paper, conclusion was made that factors such as: stereoscopic disparities, kinetic depth, texture, perspective, shadows all affect how visible the distinguishable data is for the viewer.

Based on those findings, this project is using lighting to create shadows, made in 3D, and using motion to assist in perspective of objects for the player. Believe those factors do have impact on the user experience.

## Standards for Augmented Reality: a User Experience perspective

*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.

(***REFERENCE)*** In ISO, 9241-201[1], definition of user experience is: “*a person’s perceptions and responses that result from the use or anticipated use of a product, system or service”*. With that definition, in the paper the user experience, UX, includes: responses that are either physical or psychological, preferences, user emotions, beliefs, perceptions, before/during/after behaviour and accomplishments, as well as the media that are used in the application.

In this research paper, different inputs and outputs were analysed where at the end it provides clear conclusion on what works well for the AR or VR application. However, the inputs and outputs used in the application are affected by the application goals and uses.

Inputs for AR applications that were taken into the account were: tactile, visual, auditory, sensory modalities, kinaesthetic. *Table 1* represents each input with the effects on the UX.

*Table 1: AR Inputs and how it affects UX.*

|  |  |  |
| --- | --- | --- |
| Input | Input use | Effect on UX |
| **Tactile** | * Joystick * Keyboard * Touch-screen * Mouse | * Ease of use * Accuracy * Responsiveness * Comfort |
| **Visual** | * Camera Tracking   + Marker   + Non-Marker | * Ease of use * Responsiveness * Accuracy * Privacy/Social comfort |
| **Auditory** | * Inferring user context   + Detected frequencies   + Ambient noise level   + Patterns * Voice command | * Accuracy * Social comfort * Responsiveness |
| **Sensory modalities** | * Sensors   + Detect user/environment   + Hybrid-sensors involving more than 1 sensor | * Accuracy * Seamless switching |
| **Kinaesthetic** | * Motion tracking * Track body | * Ease of use * Accuracy * Responsiveness |

Based on those findings, there are common features that affect that UX, which are namely the accuracy, responsiveness, and the ease of use. *Table 2* represents different outputs that can be used in AR applications.

*Table 2: Outputs for AR Applications*

|  |  |
| --- | --- |
| Visual | * Uses   + Contrast   + Field of View   + Brightness   + Depth perception * Content Quality   + Realism   + Abstraction   + Frame rates * World consistency and stability   + Registration   + Stability * Most important for UX * Can be simple or complex * Implications   + Health and Safety     - HMD (Headsets)       * Narrow field of view       * Inadequate depth perception       * Low display brightness       * Poor ergonomics   + Context Awareness     - Can lower implication with positional stability that is easier to accomplish |
| **Auditory** | * Directional output (speakers) * Simple to implement * Useful for immersion * Implications   + Disruptive to environment   + Health and Safety   + Privacy |
| **Haptic** | * Limited use * Focuses on touch and feel feedback * Use for Force-feedback systems   + Medical applications   + Game controllers |

From *table 2*, it can be concluded that visual output is most important aspect of UX, as it is what users interact with in the AR application. Auditory is secondary impact for UX, as it does help with immersion and to provide a sense of atmosphere, however it is limited to directional output and it may be disruptive to others. Haptic output is most limited as it uses touch and feel feedback, and for this project it will not be used, as it is hard to accomplish on mobile device.

In conclusion, from this research, the features that are focused on this project are visual aspects and auditory, to combine them to influence UX. It is accomplished by creating output that is visually stable, and satisfying field of view. Auditory is being used in helping to achieve immersion with the world that is presented using visual output.

## Existing Games

One aspect that had to be considered when implementing and working on this project was how existing titles are implemented and what makes them successful.

*Table 3: Existing titles comparison table.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Game Name | UI | | Indicators | | Features |
| AR Dragon | Menus hidden within few onscreen buttons, once clicked it shows more options.      Clean design, only buttons showing on lower sides of the screen with expandable menus. | * UI has striking colour buttons for easily distinguishable buttons. * World is spawn around you, only way to find out if to look around or if in practise mode (train mode) then the dragon looks in that direction. * Distinguishable colours when it comes to pickups, shooting targets and mood of dragon can be seen by the face expression. | | * Dragon needs to be spawn on big enough surface. * Game uses whole world around you, even outside the scanned surface. Can be seen when in train mode, or point spawn. * As Dragon grows, it requires bigger space. * Can be picked up and placed anywhere (has to be scanned first and a plane) and dragon can only fly within this plane. | |
| Space Invaders – not released | Non-Existent, very minimalistic.    Only information on screen is how many bullets are left in the gun. It is not released game, just prototype. | * + - Pixelated enemies coming towards you from every side, the only way of seeing them is by looking around.     - Models becoming bigger the closer they are to the player. | | * + - Takes advantage of the whole environment around player by spawning enemies coming at the player.     - Tap the screen to play the game (Shoot) * Tracks player location. | |
| Space Shooter – not released | Non-Existent, prototype build, minimalistic information shown on screen. | * + - No real indicators showing ships – just models floating around player. Have to look around to spot them. * Not interactive models, stand still | | * Ships spawning at random location around player * Doesn’t use planes to spawn objects | |
| ARise | Minimalistic UI – Only options and menu buttons at the lower corner.    Simplistic UI, using camera positioning to progress with the game. Fits well with the game mechanic, to have most area to display the game. | * Symbols that connect together and shine once the camera joins the symbols together. * Symbol shine up when connected and the path is created. * Use of colours to clearly show progression, player location (by using distinguishable colours compared to whole game), symbols, can see whole level at once. | | * Requires a plane to spawn the world. * Uses camera position to play the game. * Able to see whole level once loaded, allows player to plan ahead. * One spawn location, but by the game mechanic (of progression with camera position), forces people to move around the map. | |
| Splitter Critters | Minimalistic UI – Pause and restart button. Clean UI, slide screen to play the game. | * + - Colours used to differentiate different game objects. * Different texture symbolising cut. | | * Box appears with a cut out which is playable window. * Slide to cut and move objects. * Able to go in the box to see different layers of game. * Bright colours for enjoyable visual experience. | |

Titles in *Table 3* were chosen as they look similar to the game that is being implemented in this project. Common features that are implemented in those titles are: bright and distinguishable colours, simplistic and clear UI, clear game objectives and interaction. In the implementation of this project, those features were taken into the account, affecting how game was implemented. As by looking at existing titles, it illustrates that those are features that work well and provide satisfying UX.

## Summary

By completing this research, in summary, it did have an effect on how the project is being approached and is being implemented. From the research, the visual feedback for the user is very important, followed by the sound and the UI, all of which were important features that are focused on in the project. Also, using 3D objects with potential motion for greater recognition for human eye. Using bright colours, to distinguish game world from the real world, combining it with the auditory output for best possible UX. Apart from the output conclusion, the input was also considered, however due to the nature of this AR game and analysis the input options from the research paper, it did make most sense to implement visual and tactile inputs. Combining this input combination, with the outputs, and comparing existing titles, it should provide AR game project with satisfying UX.

# Method of Approach

## Approach

The production of the prototypes and tests will follow a Scrum, 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.

Reasoning for using Scrum, Agile methodology is the organisation that it provides in the amount of time provided. It offers sprints which have clear goal of what is being achieved, with an option to adjust after each sprint. Having organised sprints, with specified deadlines for each sprint, does help with limited time that this project is under, and it sets clear goals before start of implementation of what should be a focus. Another reason for choosing scrum, agile methodology is familiarity.

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

AR game can be implemented using variety different SDK, Kudan, Vuforia, ARCore, and many others, however since goal of this project was to implement good UX experience, focus was on creating a game without a marker. At this moment, ARKit and ARCore offer top quality tracking, and due to having access to only iPhone 7 device, ARKit was chosen as a technology to implement with. Other AR developer kits require to be implemented with a marker, and the game experience is limited due to the marker.

With chosen technology, there were requirements to implement with ARKit, those are reasons for implementing on macOS, having MacBook Pro available to implement on, and iMac in classroom as a back-up. Xcode was required to build on and test project on iPhone device, and using ARKit Unity toolkit to build the game in Unity. Chosen Unity due to familiarity with C# and Unity game engine. Other option was to implement in Xcode, however learning Swift would be required, and with the time limitation there was not enough time to learn it to complete project to high quality.

Adobe Photoshop was used for graphical assets, chosen for familiarity and availability.

## 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.

*More details, reasons why im using this approach, show understanding of other management approaches and why I picked those ones*

# Legal, Social, Ethical and Professional Issues

*Copyright*

*“The Copyright, Designs and Patents Act 1988, is the current UK copyright law. It gives the creators of literary, dramatic, musical and artistic works the right to control the ways in which their material may be used. The rights cover: Broadcast and public performance, copying, adapting, issuing, renting and lending copies to the public. In many cases, the creator will also have the right to be identified as the author and to object to distortions of his work.” (****REFERENCE)***

Assets, sounds, and anything else that is used in the project is protected 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.

Copyrights for items used:

*Planets and projectiles* – Unity Asset Store states that assets that were taken from the Asset Store are able to be used in personal or non-commercial purposes, some even agree to be used for commercial purposes. (*REFERENCE)*

*Sound, SFX* – Sound used in this project contain music for healthy and infected planet and projectile shot sound. All of the sounds are agreed to be used for non-commercial purposes. *(REFERENCES)*

*Code for mesh deformation* – Able to distribute the code from the site for personal, non-commercial, commercial purposes. (*REFERENCES)*

Since the Assets, Sounds, Logos and any other content created by others, are not used for commercial use, no copyright laws were broken in developing this project.

*Ethical*

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.

*Social and Professional*

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.

# Implementation

## 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.

*The following show the outcomes after each sprint:*

First sprint outcome, which involved testing the technology out, was to carry out research, which might affect the development of the project. There were no delays in this sprint and everything was working as intended.

Second outcome involved implementing prototype of flying planets, with an option for a character to jump between them. Sprint was successful, completing the goal that the sprint set out to have, with the feedback to consider depth perception, potentially shaders, and sound to represent the state of the game. After this sprint, there were no changes made to following sprints.

Third sprint outcome was the planet system with infection spreading across planets. It had a solid line between, where the feedback to make it more dynamic, for person to be able to see the progression of infection. Also, this is first time where the game orientation had to be taken into the account, and testing for the orientation was added for the next sprint to identify which orientation the game should be based on – landscape or portrait.

Fourth sprint involved completing the prototype to be ready for the testing to be completed on the next sprint. The outcome involved orientation testing to be completed and test results indicating for the game to be implemented in landscape position. This test was done by giving game to people to try out and monitor in what orientation they play the game. Also, all menus were implemented for the game, so test participants had a play-able game.

Fifth sprint involved to have testing complete, followed by adjusting sprints after to take into the account the feedback obtained at the testing session. Outcome for this sprint was that the game involved too many menus, with players taking extra, unnecessary, steps to do an action. Also, lack of sound impacted on the enjoyment of the game. Based on the feedback, the game was re-created, adjusting following sprints to involve cleaner UI, and researching more on the UX and AR, to implement more enjoyable UX experience for the player, while keeping the game mechanics the same. (*REFERENCE to appendix test session)*

Sixth sprint was improvements implementation towards the game. Outcome was more positive, compared to the initial implementation, having more enjoyable UX experience, while at the same having better gameplay. Feedback gathered was taken into the account taken from the testing session and was implemented in this sprint.

## Requirements

Minimum software and hardware requirement:

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

Project requirements:

For the project, the requirement is to create enjoyable UX experience. That involves creating visually pleasing and easy to understand gameplay. This includes information that is being presented to the player in understandable and logical manner. Goal being for the player to have a feeling of accomplishment, enjoyment, when playing the game and after the game. This is achieved by the way that UI is implemented, the sounds, and the game visuals. For this project, the goal is to implement clean, minimalistic, UI, with good introduction towards game mechanics, combining with well-designed visuals and sounds.

## 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 (*REFERENCE)*, 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.

## 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.

### 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.

### 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.

### 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.

## 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.

### 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. Existing titles comparison table is visible in section *3.3*.

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.

*ADD SCREENSHOTS*

### 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.

Initially, the game was implemented with introduction text at the beginning of the game, where feedback was not that positive towards that approach *(PROVIDE EVIDENCE)*. Upon that feedback, the design of the game was altered, to introduce the mechanics to the player in more interactive manner, which did gather more positive feedback *(EVIDENCE)*, and that is the direction that the game was implemented further.

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.

### 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.

# 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.

## 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.

## 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.

## 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.

## 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.

## 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.

## 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.

# 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.

# Project Evaluations

## End Project Report

After the project had reached its conclusion, AR game has been created which satisfies most of the project objectives that were laid out in the Project Initiation Document (PID). AR game is created which illustrates the system possibilities, with satisfying UX. The implementation includes clean, simple UI system, with the optional introduction which introduces a player to the game mechanics. Actions in the game are logical, and simple, with visually pleasing and well-defined depth perspective, for the viewer to distinguish different game objects.

The project was kept to the original objectives, without any major alterations. Within the project, how the game is to be implemented was adjusted to fit the project objective, but the core of the project stayed the same throughout the implementation. The expectations, or possibilities of AR technology might have been tested further, as this project focuses more towards UX in AR rather than the possibilities with AR system. However, possibilities are still presented, by how the game is implemented, with marker-less implementation and game interactions.

At the initial stages of the project, more research could have been done, when the implementation was being organised into sprints. This would allow more time to implement a more polished out game, as the game implementation was adjusted, changed, after week five to fit the objective more. This has occurred due to lack of research completed beforehand, and miss-understanding of what would work well in this system. Even with those issues, the project still manages to fit the objective criteria and completed a product that meets objectives.

The project was implemented, like intended, on macOS, for iPhone, iOS devices. Unity and C# was used to implement the game, by using ARKit Unity Plugin to convert the game to ARKit game and install it using Xcode on the iPhone device. This was planned initially in the PID document and did not have to change throughout the project.

## Project Port Mortem

Overall, the project was completed successfully, meeting project objectives, without any major step-backs. The use of agile methodology was used in advantage for this project, to keep the progress organised and also manage the workload throughout. With the organisation of sprints, there was a clear goal for the sprint that was going to be achieved, which helped with the overall progress of the project. The software and technology usage was also successful, even though time was spent at the beginning of the project dedicated mainly on learning how to operate with the ARKit technology.

Even though the project met the objectives, there were some issues along a way that could have been resolved if organisation was better at the start of the project. The project implementation was changed in the middle of the implementation, which was due to lack of research, and lack of game planning. That created a situation where the prototype was implemented with incorrect features in it, causing the whole project not to meet the objective. How it could have been avoided is by having done proper research and plan the application before the implementation, creating prototypes and gather feedback. This would delay the start implementation a little bit, but overall the time spend on the implementation would have been longer, as in this project some implementation time was wasted due to change of implementation in the mid-way of the project, once the research was completed.

There were no issues with the software, apart from learning ARKit at the beginning of the project, however the organisation took that into the account, by having simpler mechanics and features to be implemented initially. This allowed time to learn ARKit toolkit in Unity. It allowed the project to be implemented in the Unity game engine.

Organisation was well implemented, apart from the initial research, as sprints were planned out for the whole project, keeping track on the management tool. Also, developer diary was kept up to date, which had entries whenever any implementation took place, which was useful in referring back on the implementation stages and decision choices that were made.

## Objective Evaluations

Objectives in the projects were met, and evaluated by the feedback received when testing the game. Main objective for the project was in creating AR game application, with well implemented UX. This involved having game that interacted with the player, creating enjoyable experience for the player. This was achieved by the way that UI is implemented, and how player interacts with the game that is in logical manner. Visuals and sounds are also used to assist good UX implementation.

For this project, clean UI, with tap screen was implemented for interaction with the game. Visual particles, and assets were used to show the game state to the player, using distinguishable colours and particles to represent the game. Sound was also used in assisting to show the state of the game to the player. Game mechanics are presented to the player one by one, in visual form, which introduces the game to the player gradually, one aspect that is important in good UX.

Combining all of those features, the game implemented is believed to contain suitable UX for the application created. This could be seen from the two test cases that were completed, one prototype showing badly implemented UX, and final result which contain well implemented UX system. *REFERENCE TO APPENDIX.* Having positive feedback gathered for the AR game, suggests that the project has successfully completed the project objectives.

# Conclusions

## Lessons Learnt

This project was a learning experience, improving knowledge with programming in C#, and using Unity game engine, having minimal experience in using C# and Unity before. Application and software development skills were also improved, as this is project where application, in this project a game, is created from design stage towards a functional game.

Apart from improving current skills, new skills and knowledge was learned, working with ARKit toolkit to implement game in AR, having no prior experience or knowledge with the technology beforehand. Moreover, focusing project on UX implementation, which required research and improving UX knowledge in creating a successful game, which required to learn more about UX.

With the strict deadlines, this project also required good organisation skills, and required to be have good time management. Both of which were improved when working on this project, keeping to the deadlines that were set throughout the project for the project deadlines, but also the sprint deadlines.

Overall, this project did offer many opportunities to learn new skills, which was highly enjoyable. Apart from learning new things, improving on the current skillset is also important and it was also accomplished in this project, and it was required for the project to achieve the set objectives.

## Further Development

If this project was being continued further, there would be some changes which were sacrificed due to time limitations. Primary further development that would take place would be to increase level complexity, as at the moment there is only one level that is available to play. Following the level implementation, the progression system would be implemented, which involves unlocking abilities with levels completed. With the progression, the rewards system would also be implemented, in the form of a star system as many other mobile games use.

Apart from adding features and mechanics, some things that would be changed are the assets used in the game, which suit a theme, with shader interaction when the player clicks on the planet instead of the mesh deformation. Also, menu system would be implemented with full features, and graphically pleasing. Removing all bugs would also be a priority in further development of this project.

## Conclusion

In conclusion, this project was successful, with well-designed UX in AR game. This was successful with well organised project milestones, using scrum, agile methodology, which organised project with milestones to meet. Testing and gathering feedback was also important, as it did affect the implementation of the game. To avoid issues in this project, research would be completed earlier, to avoid miss-understanding on the project objectives. This project did provide insights into UX and AR game development.

# References

To be completed…

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

1. S. L. Ames, J. S. Wolffgohn, and N. A. McBrien. The development of a symptom questionnaire for assessing Virtual Reality viewing using a head-mounted display. *Op- tometry and Vision Science*, 82(3):168–176, March 2005.

# 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