
# AUGMENTED REALITY WHACK-A-MOLE

April 2018

Jasmine Dodson
Manchester Metropolitan University
School of Science and Engineering
Computer Games Technology
Supervised by Kevin Tan

## DECLARATION

No part of this project has been submitted in support of an application for any other degree or qualification at this or any other institute of learning. Apart from those parts containing citations to the work of others, this project is my own unaided work.

Signed: Jasmine Dodson


## ACKNOWLEDGEMENTS

I would like to thank Dr Kevin Tan for his constant support throughout this project.


## ABSTRACT


With the increase in popularity of both virtual and augmented reality in video games, especially for mobile devices, as well as the rising popularity of 'on-the-go' games, the Augmented Reality 'Whack-A-Mole' project looks to bring the two concepts together to create a new video game for mobile devices. The game targets those interested in augmented reality for games, as well as 'casual' gamers. The final project of this report is a playable game with a custom user interface, with some art assets not included.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES



# LIST OF CODE



# ABBREVIATIONS

AR – Augmented Reality

UE4 – Unreal Engine 4

UI – User Interface

Unity – Unity 5

VR – Virtual Reality

# 1 INTRODUCTION

## 1.1 PROJECT BACKGROUND

> *People love to play, and technology is an arbiter, an umpire, a scorekeeper, and a dungeon master all in a really great way.*
>
> Nolan Bushnell (Melissinos and O'Rourke, p25, 2012)

Released in 1972 by Atari, the now classic coin-operated arcade video game *PONG* helped kick-start the arcade gaming industry, alongside *Computer Space*, which Atari had released the year before (Kent, 2001). The eighties saw a rise of dedicated video game arcades (Langshaw, 2011), with familiar titles like *Galaxian*, *Pac-Man*, and *Donkey Kong* added to the roster of available games. Taito's *Space Invaders* released in 1978 was an immediate success and converted to a home system in 1980 (Melissinos and O'Rourke, 2012). Thus, slowly began the decline of the classic arcade games, while the home console would rise to levels of popularity it retains today.

The video game industry continues to boom, with digital games accounting for 87% of the global market, generating $94.4 billion out of an expected $108.9 billion as of April 2017 (McDonald, 2017). The mobile games industry itself has grown 19% and the prediction for April 2017 was that it would claim 42% of the market; by the fourth quarter, mobile games held 43% of the market share (Newzoo, 2017). This continuous rise makes it a worthwhile industry to develop for, especially as "*mobile phones are a thing that just about everyone has on them at all times these days. This provides players with the opportunity to play games anywhere they want, just by reaching for the phones, for the most part regardless of their present situation*" (Crews, 2017).

Augmented reality and virtual reality themselves are also on the rise, in part thanks to the rise in their use in video games both for educational and entertainment purposes. "VR and Augmented Reality (AR) would in the long term change the way consumers communicate and interact with each other and content" (Bhat, 2017) as evidenced by smartphone applications, covered in this report, such as the popular augmented reality game Pokémon Go, that places 3D models of the creatures into the real world to be captured.

This project aims to not only digitise the game *Whac-a-Mole*, but do so in augmented reality for a smartphone, to take advantage of how readily available smartphones are, as well as the novelty of augmented reality as a unique feature of the game.

## 1.2 AIM

The project aims to design an augmented reality video game similar to the popular arcade game 'Whack-A-Mole' which can be implemented primarily on mobile devices, that involves the development and implementation of an Augmented Reality (AR) framework that can be used to display creatures from the game.

## 1.3 OBJECTIVES

### 1.3.1 Research

- Existing augmented reality software
- Differences in Unity and Unreal Engine for use in AR game development
- Augmented reality and its use in relation to video games
- Computer Vision

### 1.3.2 Development

- Game mechanics
- Augmented reality framework – must work with at least one 3D model
- Animations for use with augmented reality
- User interface

### 1.3.3 Testing and Feedback

- Survey on the app and game as a whole
- Feedback on usability of the application
- AR framework tested on mobile device

# 2 LITERATURE REVIEW

## 2.1 COMPUTER VISION

### 2.1.1 How It Works

Computer vision is the “*technical capability that enables a computer to recognize or understand the content of a picture*” (Lerner and Lerner, 2013). It tries to replicate “*what a human brain does with the retinal data*” (Sharifzadeh, 2017), which includes machine learning and image processing to analyse image content.

Images captured by a camera go through pre-processing where parts of it deemed unimportant are discarded before the computer can “*analyze the information by identifying and defining any objects that may be significant*” (Lerner and Lerner, 2013). One of the key operations of pre-processing is edge detection where “*areas of the image that possess regions of clearly differing light intensity*” (Lerner and Lerner, 2013) are identified to begin defining objects.

### 2.1.2 Quick Response Codes

Quick Response codes, more commonly referred to as QR codes, made possible through advances in computer vision, are two-dimensional patterns that work in a similar way to the one-dimensional barcode. The code “*carries information both horizontally and vertically* […] [and] *is capable of encoding the same amount of data in approximately one-tenth the space of a traditional bar code*” (QR Code Features, 2013).

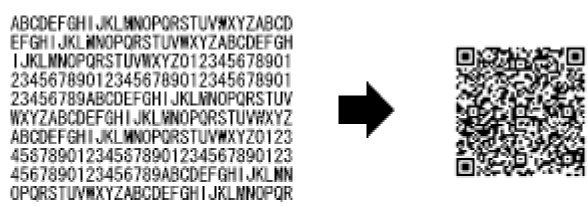



*Figure 2.1 - QR Code symbol encoding 300 alphanumeric characters. (Denso Wave Incorporated, 2013)*

Where regular barcodes are scanned mechanically by reading light bounced back from the white gaps, QR codes can be read digitally by a smartphone's camera and interpreted by the system (Tarantola, 2012). Uses for QR codes include “*everything from inventory tracking, to shipping and logistics, to online ticketing*” (Tarantola, 2012).

### 2.1.3 Pattern Recognition

Pattern recognition deals with the "*description and classification of measurements taken from physical or mental processes*" (Fu and Rosenfeld, 1976), and is the "*process of classifying input data into objects or classes based on key features*" (Pattern Recognition, n.d.). Feature detection and extraction are both used in order to rapidly reduce an image down that "*efficiently represents interesting parts of an image as a compact feature vector*" (Feature Extraction, n.d.) for pattern recognition.

Supervised and unsupervised classification are the two methods used for pattern recognition. Supervised classification uses algorithms to assign labels from training data that contains what the machine should look for. This method is used for object detection, face detection, and face recognition (Pattern Recognition, n.d.).

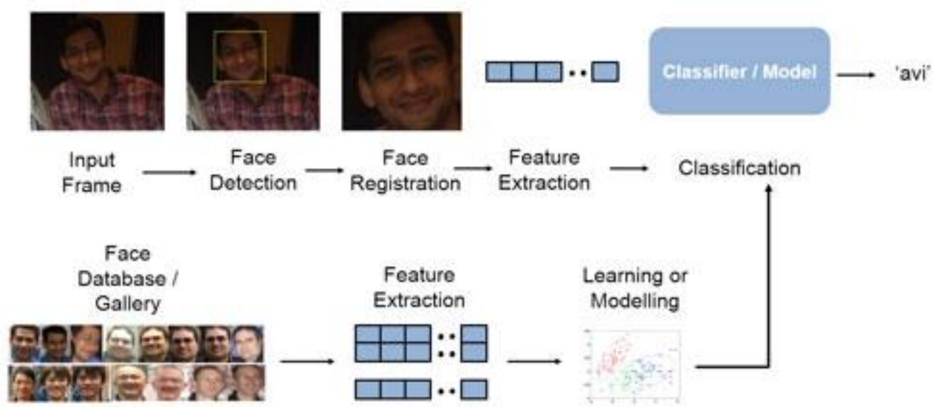



*Figure 2.2 - Steps in the face recognition workflow (The MathWorks, Inc, u.d.)*

For unsupervised classification, this method works by "*finding hidden structures in unlabeled data using segmentation or clustering techniques*" (Pattern Recognition, n.d.) and is used for object detection as well as image segmentation. Methods used for this classification include: K-means clustering, Gaussian mixture models, and hidden Markov models (Pattern Recognition, n.d.).

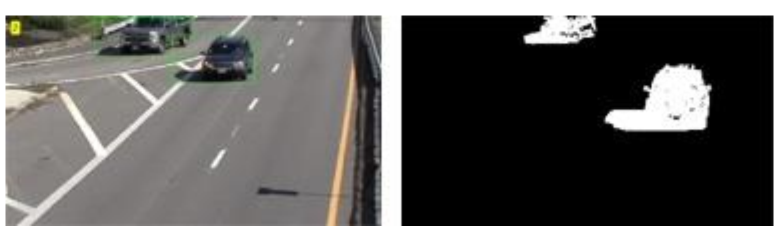


*Figure 2.3 - Gaussian mixture models (The MathWorks, Inc, u.d.)*

## 2.2 WHAT IS AUGMENTED REALITY?

Augmented Reality, also referred to as AR, is the "*integration of digital information with the user's environment in real time*" (Rouse, 2016). The digital information has "*a physical space or location in the real world*" (Craig, 2013) and the new, digital, information is simply displayed over the physical world. Augmented reality therefore has a wide range of applications for a rapidly digitised world, including, but not limited to, medical, travel, entertainment, and education.

### 2.2.1 Differences in Augmented Reality and Virtual Reality

While both augmented reality and virtual reality refer to similar concepts of changing the way the world is perceived, the terms carry slight differences.

Virtual reality is a "*computer-generated simulation or recreation of a real life environment*" (Virtual Reality vs. Augmented Reality, 2015) often viewed through headsets to view an entirely digital world. Virtual reality has become more accessible to the wider public through headsets for instance: Facebook's Oculus Rift (Oculus, 2017), and Sony's PlayStation VR; it is used for entertainment purposes like movies, or video games like Bethesda's Skyrim VR. It is also being used for simulating environments to enhance training and has applications in healthcare and defence.

*Figure 2.4 - Skyrim VR (Bethesda Softworks LLC, 2017)*

*Figure 2.5 - Soldiers using virtual reality for training (Wareable, 2015)*

Augmented reality, in contrast, "*enhances experiences by adding virtual components* […] *as a new layer of interaction with the real world*" (Virtual Reality vs. Augmented Reality, 2015) and is displayed through mobile devices such as smartphones and tablets. It can also be viewed through headsets in the same manner as virtual reality, the key difference is that augmented reality is an overlay of information in the user's field of view.


*Figure 2.6 - Nokia City Lens' augmented reality mode (Wired, 2012)*

### 2.2.2 Mixed Reality Spectrum

"*Mixed-reality systems* […] *combine the power of computers with the complexity of the physical world*" (Mixed Reality, 2013). Augmented reality "*can be regarded in terms of a continuum relating purely virtual environments to purely real environments*" (Milgram et al., 1994). Paul Milgram's concept on the reality-virtuality continuum defines augmented reality as the real world being augmented by the virtual; in contrast augmented virtuality is defined as the virtual world augmented by the real.

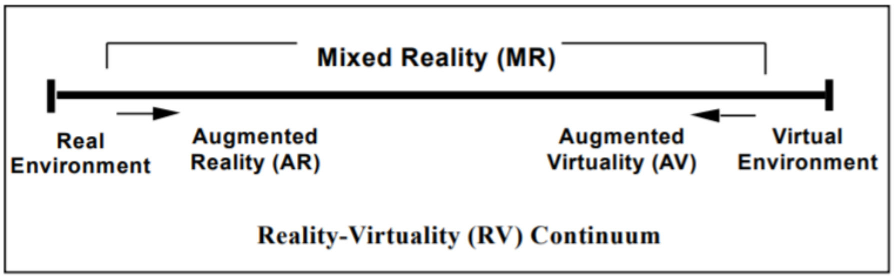



*Figure 2.7 - The reality-virtuality continuum (Milgram et al., 1994)*

### 2.2.3 How It Works

The hardware required for augmented reality can be broken down into three core elements: a display such as a headset like Google Glass, a mobile device, or any other type of screen; sensors for spatial registration in order to place virtual objects correctly in the world, especially in the case of mobile augmented reality where devices are more easily moved and rotated which could affect the orientation of the virtual object; and a processor capable of rendering the virtual overlaid on the real, as well as processing and storing location data (Höllerer and Feiner, 2004) (Craig, 2013).

The core steps for processing augmented reality as outlined by A. Craig, 2013, are:

1. *The application needs to determine the current state of the physical world and determine the state of the virtual world*
2. *The application needs to display the virtual world in registration with the real world in a manner that will cause the participant(s) to sense the virtual world elements are part of his or her physical world and then return to step 1*

In following the above steps, sensors within the device track the six degrees of freedom, such that, regardless of device location or orientation, the virtual object is still in its expected position if it is a static object, like a bottle. For virtual objects capable of movement, for example, certain video game elements, the location for the next frame is relative to the location of the participant.

## 2.3 EXISTING AUGMENTED REALITY SOFTWARE

### 2.3.1 Snapchat

Social media has expanded to include augmented reality within camera features, with Instagram (Systrom and Krieger, 2010), and Snapchat (Spiegel et al., 2011) having millions of regular users. Snapchat's 3D World Lenses allows users to "*add captions, color, characters* […] *to the world around*" (Snapchat Support, 2017) them, including the ability to resize, reposition, and walk around the virtual content. With Lenses, Snapchat have shown the potential for companies, with "*ad campaigns using them resulting in increased ad and brand awareness*" (Locklear, 2017).

*Figure 2.8 - Snapchat 'dancing hot dog' filter through augmented reality (USA Today, 2017)*

### 2.3.2 Ikea

The development of Ikea's augmented reality app Place in Your Room allows users to "*place virtual furniture in* [their] *own home with the help of augmented reality*" (Ikea, 2013). Through scanning catalogue pages or selecting an item through the digital collection, furniture can be previewed within a home in order for customers to find the perfect piece.

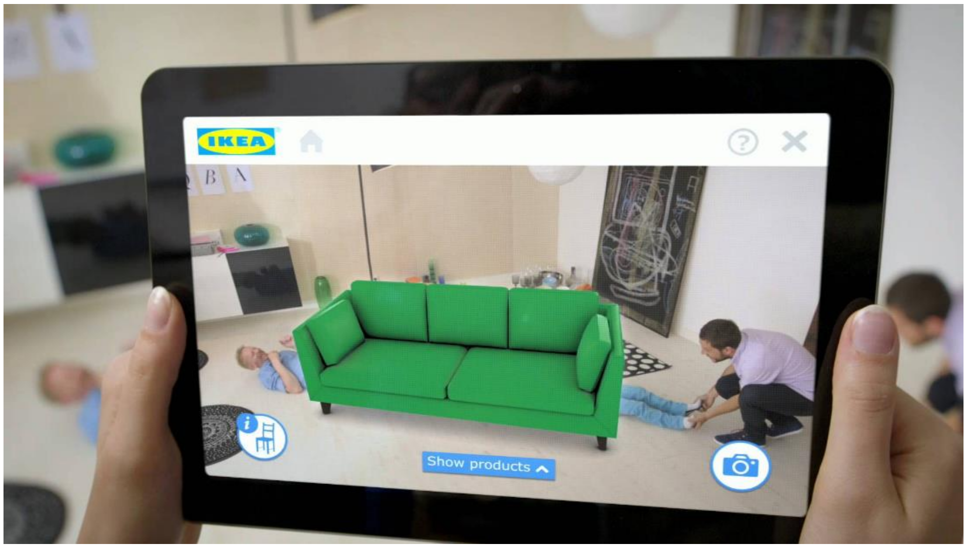



*Figure 2.9 - Ikea AR mobile app allows trialling of furniture (Ikea, 2013)*

### 2.3.3 Pokémon Go

Widely played throughout the world, Pokémon Go is the "*first mainstream case of augmented reality applied to the masses*" (How Pokemon Go Took Augmented Reality Mainstream, 2016) with the game utilising smartphone's GPS and cameras, "*the game makes it seem as if wild Pokémon are cropping up on the streets of the real world*" (Brewster, 2016), with different creatures appearing at different locations.


*Figure 2.10 - Pokémon in the real world. (The Guardian, 2016)*

## 2.4 WHACK-A-MOLE

Oxford Dictionaries defines Whack-A-Mole as an "*arcade game in which players use a mallet to hit toy moles, which appear at random, back into their holes*" (whack-a-mole, n.d.). The system has inspired various spin-offs, including a Whack a Banker game in Suffolk (BBC News - Bankers 'whacked' in arcade game, 2009).

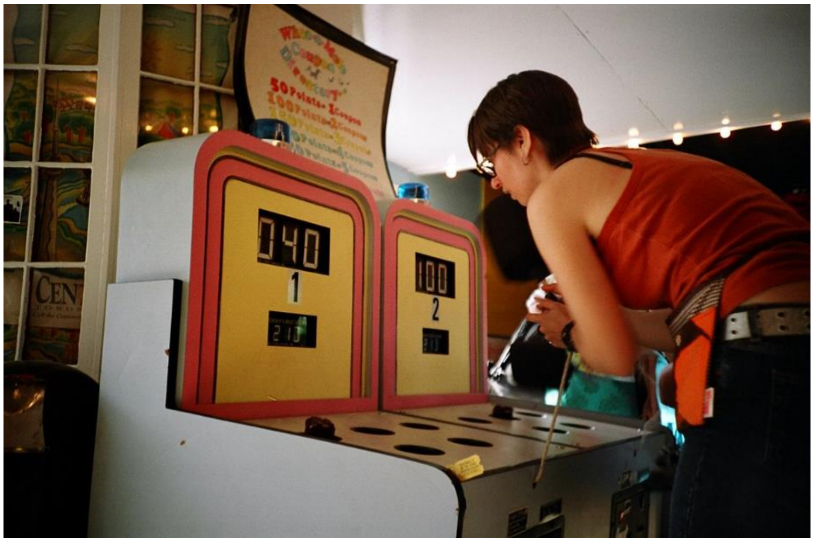



*Figure 2.11 - whack-a-mole Gayla's highscore = 140, 2005*

### 2.4.1 HoloLens Prototype

Following the *Playtest* episode from Black Mirror, David Robustelli (2017) designed an unreleased mixed reality prototype of a *Whac A Mole* game for Microsoft's HoloLens.

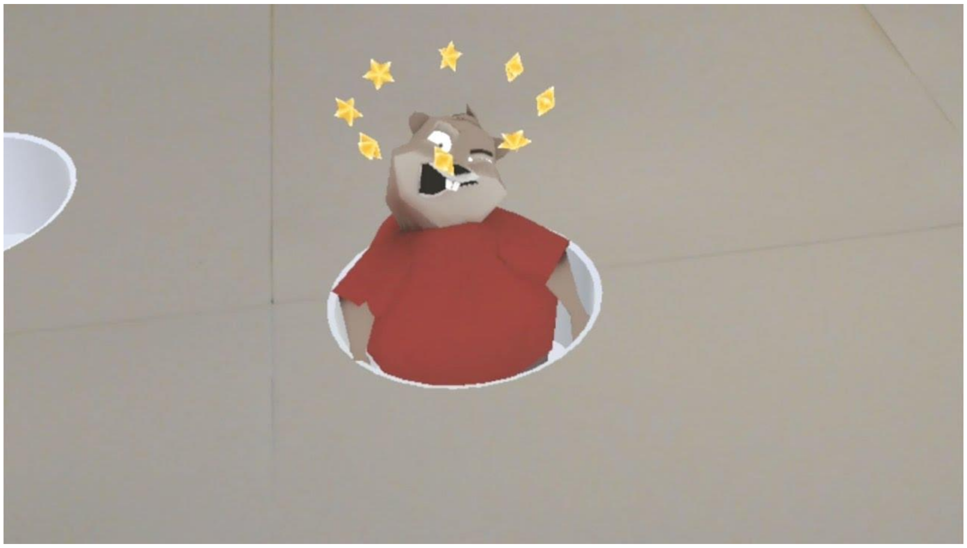


*Figure 2.12 - Whac A Mole AR/MR HoloLens prototype (Robustelli, 2017)*

## 2.5 SOFTWARE

### 2.5.1 Game Engines

Both Unity 5 and Unreal Engine 4 game engines support development for augmented reality, as well as development for mobile game applications. Game engines simplifying the development process by including either integrated tools for augmented reality development, or support for third-party plugins, as well as streamlining prototyping.

#### *2.5.1.1 Unreal Engine 4*


*Figure 2.13 - Unreal Engine Level Editor (Epic Games, Inc, n.d.)*

Unreal Engine 4, also known as UE4, is “*easy to prototype gameplay elements quickly*” (Skelly, 2016) in, as it “*makes the* [development] *process a whole lot more non-programmer friendly*” (Wong, 2017). A designer is able to make use of the Blueprint editor to visually script classes using the “*full range of concepts and tools generally only available to programmers*” (Blueprints Visual Scripting, n.d.) with the node-based interface used throughout the engine. It is “*ideal for simple in-game logic flow and sequencing of action*” (Unreal Engine 4 For Unity Developers, n.d.), and small games can be made using Blueprints alone, as opposed to having to program functionality in C++. The engine also has several “*built-in systems that you’d have to find through plugins in Unity*” (Wong, 2017) including: its own material editor; Cascade, the dedicated particle system; networking; and level streaming.

The engine does, however, have its flaws. There is a lack of supporting documentation on "*scripting and* [blueprint] *node references* [...] *you'll get the inputs and outputs of a function but they don't explain what the function does*" (Wong, 2017). Some tools in Unreal can be unfriendly to users, requiring a steep learning curve since "*Unreal tend to prefer you to work with their structure of doing things*" (Choo, 2016) and the tools can feel as if they were "*developed for an existing* [development] *team already in motion*" (Wong, 2017).

### *2.5.1.2 Unity 5*

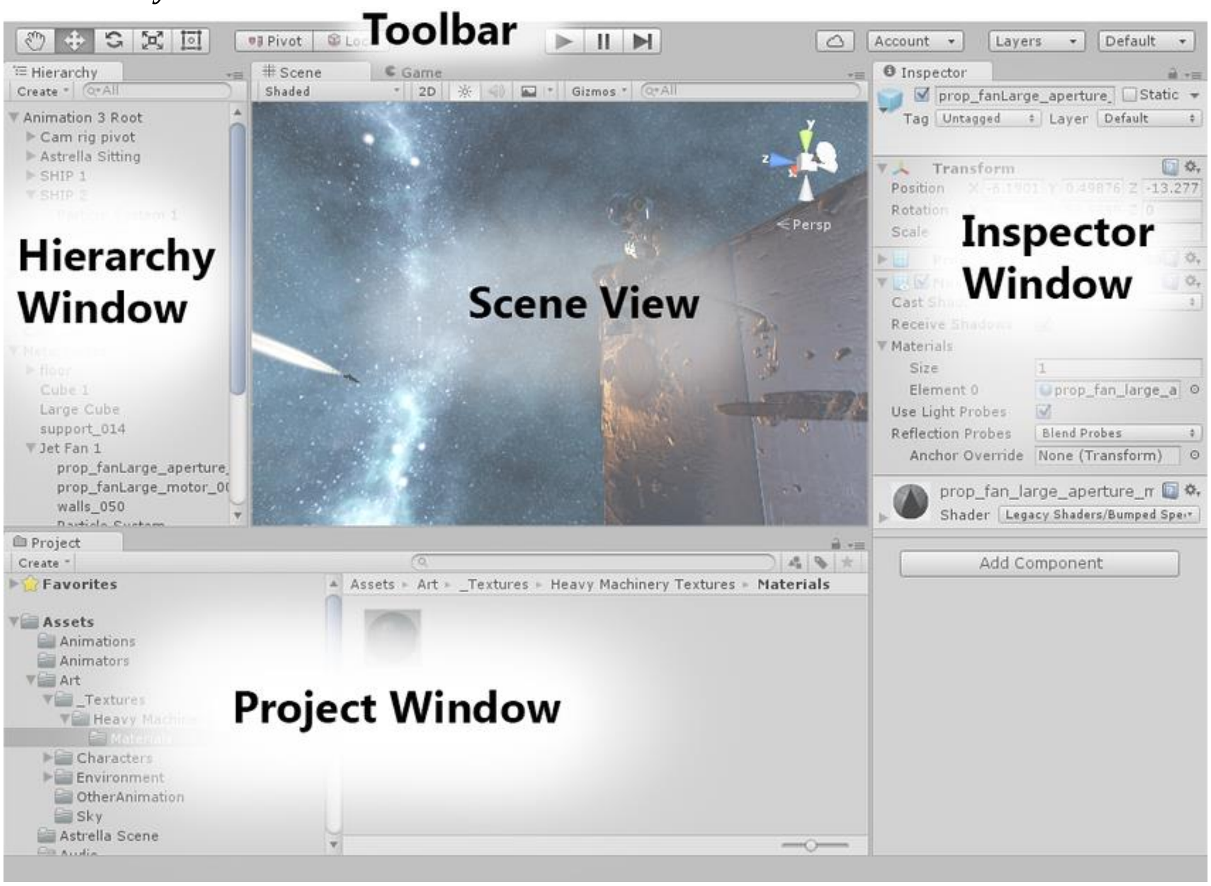



*Figure 2.14 - Unity Editor Breakdown (Unity Technologies, n.d.)*

Unity 3D 5, also referred to as Unity 5 or Unity, is an engine that uses both C# and Java for programming and currently supports more platforms than Unreal Engine 4 (Unreal Engine vs Unity 3D Games Development: What to Choose?, 2017). "*Unity 3D is fast, and the interface is quick and responsive*" (Unreal Engine vs Unity 3D Games Development: What to Choose?, 2017) with fast build times. It is also considerably "*easier to do simple tasks in code than Unreal*" (Wong, 2017). Where the supporting documentation lacks for Unreal Engine, the community for Unity is very active, "*there's a VERY good chance someone else has already tried* [something] *and posted* *about it on the internet*" (Wong, 2017) making information more readily accessible for development in Unity. Unity also has built-in support for mixed reality, including augmented reality, development, as well as a versatile API (Unity - Unity - Multiplatform - VR-AR, n.d.).

However, Unity requires significant programming knowledge which members of a development team may not have, for example "*if a designer wants to prototype something or if an artist wants to try something on their own it's extremely hard*" (Wong, 2017). Moreover, unlike engines such as Unreal which can be loosely described as open-source, "*Unity provides source code access to companies willing to shell out a lot of money for the privilege, everyone else is unable to learn and modify the internal workings of Unity*" (Knauss, 2016) which can make customising the engine for specific needs difficult for those without the funds.

### 2.5.2 Third Party APIs

#### *2.5.2.1 Apple's ARKit*

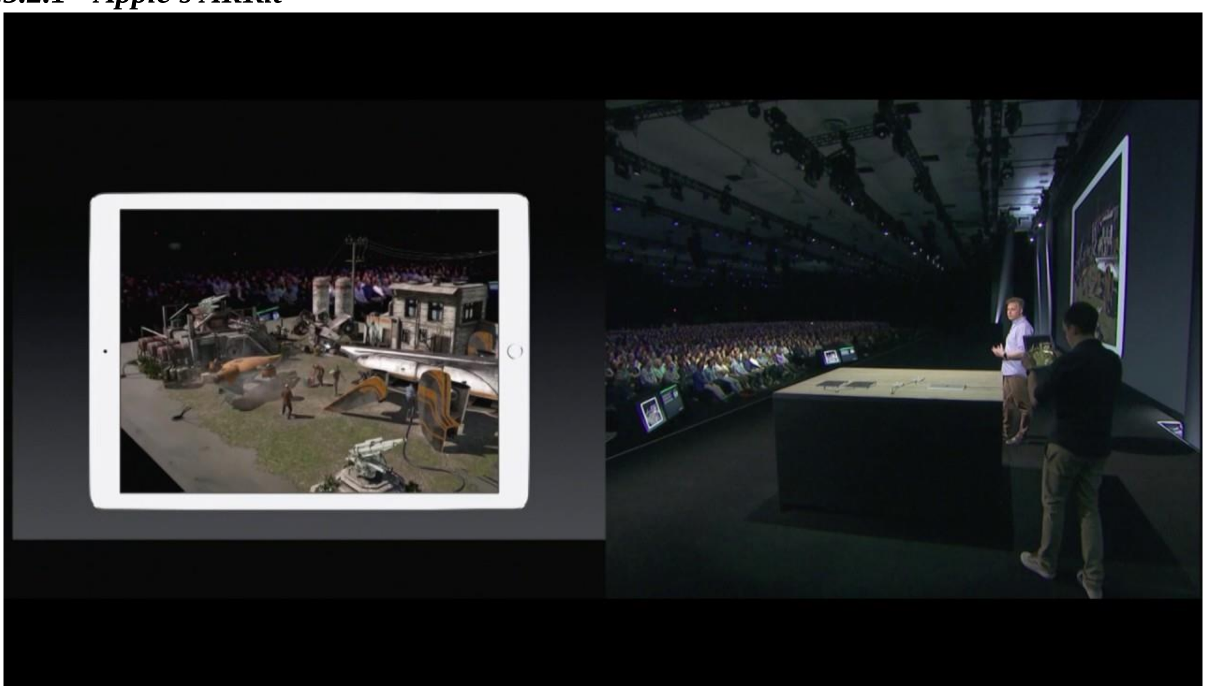


*Figure 2.15 - Wingnut AR Unreal Engine Demo on iOS | WWDC 2017 (Unreal Engine, 2017)*

For Apple's WWDC augmented reality demonstration (Unreal Engine, 2017), the Wingnut AR demonstration was created with Unreal Engine 4, alongside ARKit, Apple's "*mobile AR platform for developing augmented reality apps on iOS*" (Apple Inc, 2017). Designed to be simple for starting development, "*there's no external setup required, no pre-existing knowledge about [the] environment, as well as no additional sensors*" (Apple Inc, 2017). For tracking of the general environment, ARKit makes use of "*Visual Inertial Odometry* […] [which] *fuses camera sensor data with Core Motion data* [which allows the] *device to sense how it moves within a room with a high degree of accuracy*" (ARKit – Apple Developer, 2017). ARKit is also able to detect flat planes for object placement, as well as estimate lighting on the object from the lighting of the scene (ARKit – Apple Developer, 2017).

Support for ARKit in Unreal Engine was made available in Unreal Engine 4.17 Preview (Whiting, 2017), with Unreal Engine 4.18, released in October, adding optimisations (Cowley, 2017) and production-ready support (Wilson, 2017). For support in Unity, a plugin has been developed (Unity ARKit Plugin, 2017) compatible with Unity 5.6.2 or higher. Development for ARKit requires a mobile device compatible with iOS 11, which includes: iPhone 5S or higher, 9.7-inch iPad Pro or larger, iPod Touch 6th Generation, and more (iOS 11, 2017); it also requires a Mac running XCode 9 to package and deploy the application to the device (ARKit – Apple Developer, 2017).

#### *2.5.2.2 Google's ARCore*

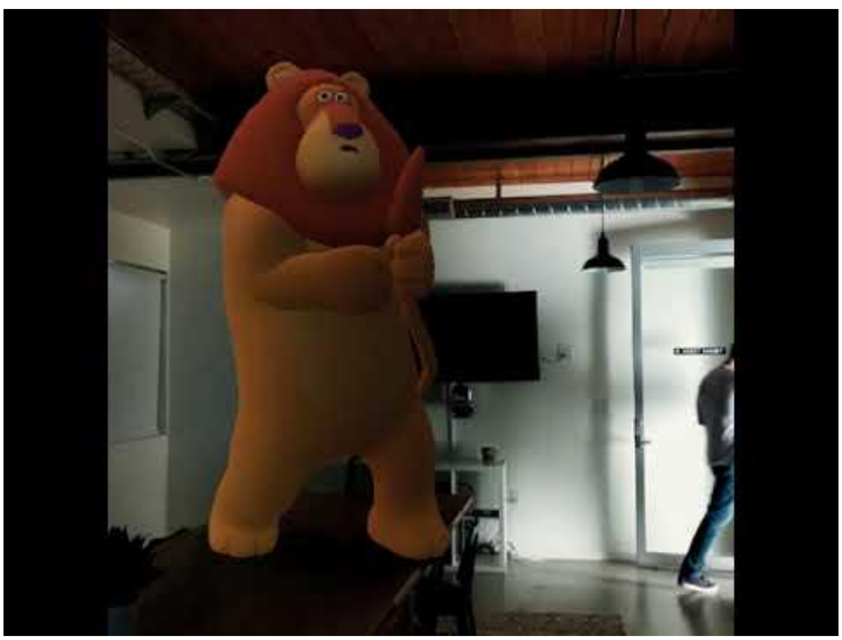


*Figure 2.16 - Google's ARCore demo Afraid of the Dark (Google VR, 2017)*

Currently being "*offered as an* ***early preview*** *so that you can start experimenting with building new AR experiences*" (ARCore Overview, 2017), Google's ARCore is designed to work on multiple devices running Android 7.0 Nougat, though currently the preview version supports Google Pixel, Pixel XL, Pixel 2, Pixel 2 XL and select models of the Samsung Galaxy S8 (ARCore

Overview, 2017). Much like Apple's ARKit, ARCore makes use of visual inertial odometry to track the movement of the camera relative to the environment that it's in. ARCore is also able to make various horizontal surfaces into flat planes for virtual objects to be placed upon, as well as estimating the lighting on the objects placed (Whiting, 2017).

From Unreal Engine 4.18 onwards, Google's ARCore developer preview is officially supported by the engine (Wilson, 2017). Unity also supports ARCore in the beta version Unity 2017.2.0b9 or any later version (Unity – Google ARCore partner page, 2017).

#### *2.5.2.3 Vuforia*


*Figure 2.17 - Vuforia Image Targets (Unity Technologies, u.d.)*

An official Unity partner, Vuforia is "*the world's most widely used platform for AR development*" (PTC Announces a Major New Release to Vuforia Augmented Reality Platform, 2017), with Vuforia 6.5 coming pre-packaged with Unity 2017.2 – the "*integration provides a number of benefits including a streamlined development workflow*" (Vuforia Developer Portal, 2017). Features of Vuforia include: digital object tracking, image recognition, and video playback over static images (AR Features, 2017). The software has been used for augmented reality applications such as Incredebooks: Disney Edition, Vuforia Chalk, and Rolls-Royce Trent 1000 Augmented Reality (App Showcase, 2017).

For mobile devices, Vuforia supports Android devices running Android 4.0.3 Ice Cream Sandwich or higher, and Apple devices running iOS 8 or higher; the Microsoft Surface Book is also supported (Supported Devices, 2017). A third-party plugin for Unreal Engine 4 was developed, however, the plugin uses Vuforia 6.2.10 (Vienna and Tappeiner, 2017) and therefore lacks functionality added to 6.5.

## 2.6 Evaluating Games

There are a number of ways to gather feedback to gauge the overall success of the augmented reality game. Some of the methods outlined below lend themselves more towards numerical data which may prove more useful at the end of this project than descriptive responses. Each of them can be used in conjunction with each other and are likely to be used at different stages of the development process as the game is play-tested by various participants at each stage.

### 2.6.1 Questionnaires

Quick to analyse, questionnaires provide a fast and mass-producible method of gathering quantifiable data (9. The advantages and disadvantages of questionnaires, n.d.). Questionnaires can also be replicated (Advantages and Disadvantages of using Questionnaires, 2012) at a later date to gauge any changes in feelings or behaviour towards the surveyed item.

There are disadvantages to using questionnaires as Nederhof (1985) identifies the two factors of self-deception and other-deception that can cause social desirability bias to affect the validity of the results. Answers most likely to be affected are those pertaining to personal details including: income; patriotism; intellectual achievements; and illegal acts; that are denied or inflated in order to make the subject appear in a more favourable light. Furthermore, it can be difficult to ensure that all respondents have understood the questions as intended and that responses are given in the context of the situation (9. The advantages and disadvantages of questionnaires, n.d.).

### 2.6.2 Focus Groups

According to Merriam-Webster (n.d.) and Oxford Dictionaries (n.d.), a focus group can be defined as a demographically diverse, small group of people representative of the larger population whose responses can be used to gather feedback or determine the general response of the population. Focus groups allow participants to discuss interviewer questions with one another and provide qualitative data, with key points of discussion noted down to be analysed later. They also provide quotes that could be used later to describe the project, as well as giving both participants and interviewer the opportunity to get clarification from each other (Focus groups, 2006).

However, focus groups are subject interview bias which can take one of three sources (interview bias, n.d.): interviewer - defined as a "*partiality towards a preconceived response based on the structure, phrasing, or tenor of questions asked in the interviewing process*" (interviewer bias, n.d.); respondent - where the participant may lie or evade questions; and the situation itself can affect the responses given. Focus groups can also be affected by unrelated discussion which wastes time and resources as the conversation is distracted from the main purpose (Focus groups, 2006).

### 2.6.3 Think Aloud

The think aloud method of usability testing as defined by Nielsen (2012) is where "*you ask test participants to use the system while continuously thinking out loud — that is, simply verbalizing their thoughts as they move through the user interface*." Think aloud testing is especially useful for "*determining users' expectations and identifying what aspects of a system are confusing*" (Think Aloud Testing, n.d.) and would be most beneficial during the early prototyping of the augmented reality portion of the project. However it is a highly flexible method of testing that can easily fit into any stage of the development process. It is also a relatively cheap method of evaluation and can be used in conjunction with other data gathering methods, as think aloud testing does not lend itself to normally producing detailed statistics.

As well as a lack of detailed feedback, think aloud testing is an unnatural situation for participants to be in, simply because "*most people don't sit and talk to themselves all day*" (Nielsen, 2012). This will require the interviewer to prompt the participant to ensure they keep talking, especially because this method is also subject to biases that can affect the responses, such as pausing to filter their statements when ideally what is required is constant talk of their experience in real-time.

# 3 DESIGN

## 3.1 PRODUCT DESIGN

### 3.1.1 Game Engine

Despite an aesthetic preference for Unreal Engine 4, it was determined that Unity 5 would be the best game engine for building the game. This was simply due to the ease of exporting to multiple devices that supported augmented reality, versus a singular Apple product or limited Android selection with an augmented reality project created in Unreal Engine.

Because of Vuforia's integration with Unity, it was the natural choice as the augmented reality API to use for this project. Vuforia also makes the process of including augmented reality into projects easy with a pre-made 'AR Camera' and image target database creation through the website (Figure 3.1).

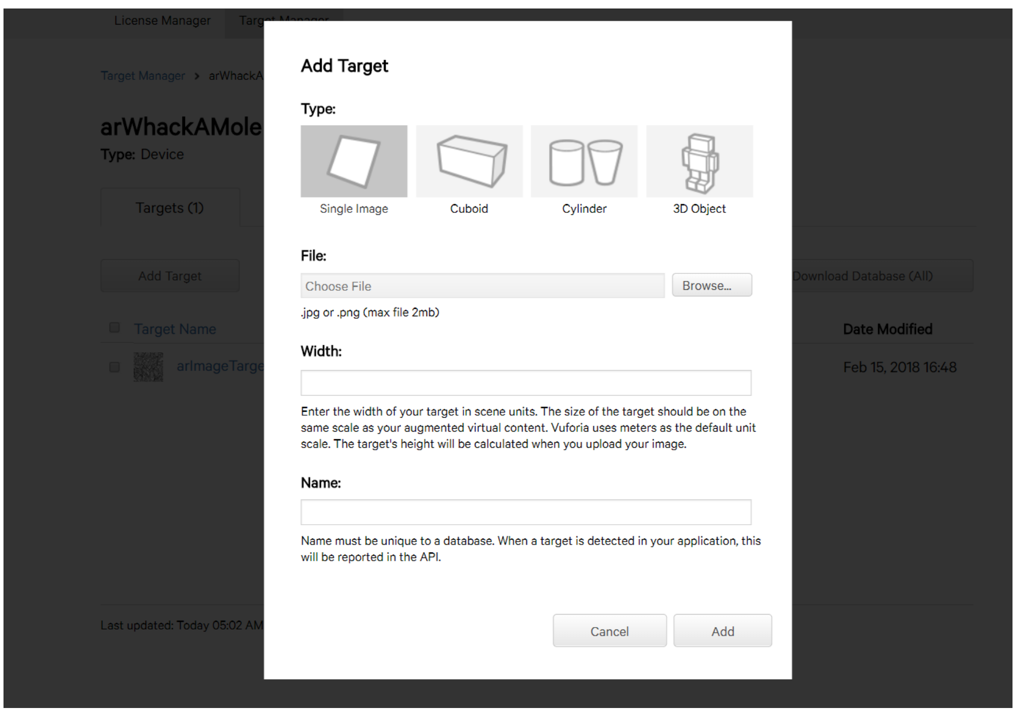



*Figure 3.1 - Image target creation made easy*

### 3.1.2 User Interface

The below set of sketches illustrates what the user interface is intended to look like for the game. The buttons within menus should be large and bright so that they are easily identifiable, as well as easy to use, as mobile phones have small screen sizes.

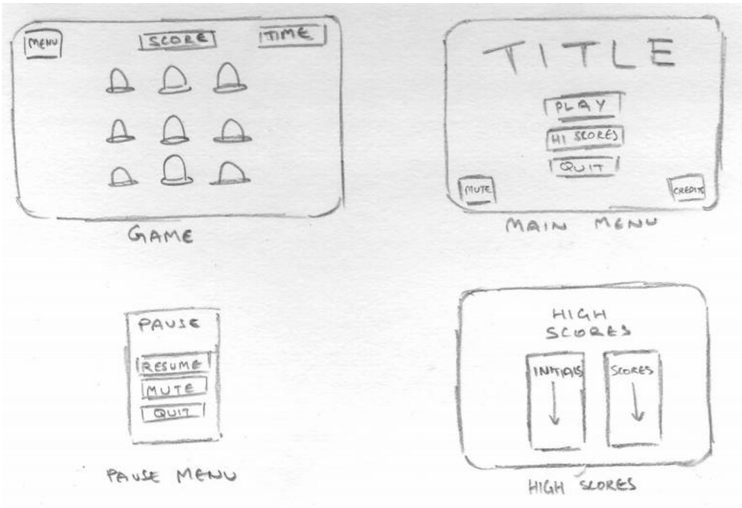



*Figure 3.2 - Interface sketches*

Figure 3.3 displays an example of a mobile game interface that illustrates the need for the buttons to have a high contrast against the background and be large enough to use easily.

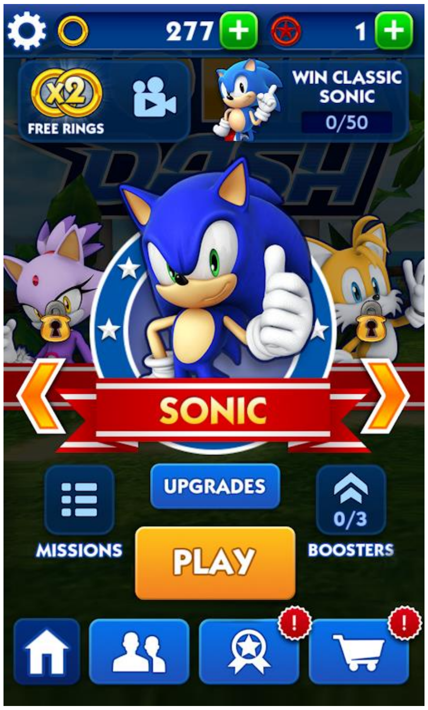



*Figure 3.3 - Sonic Dash Character Select Menu (SEGA, 2013)*

Figure 3.4 shows a brightly coloured button that would be large enough for any button text, making the interface cohesive. Smaller buttons, such as that for the Mute (Figure 3.4) use the same colours in a smaller format and use icons in place of text. Figure 3.5 shows the background used for the menus, again following the same minimal vector style as the buttons, using muted blue colours to create more contrast with the buttons. The menu background image has three moles positioned near the trees to fit with the 'Whack-a-Mole' title of the game.

*Figure 3.4 - Buttons in the user interface*

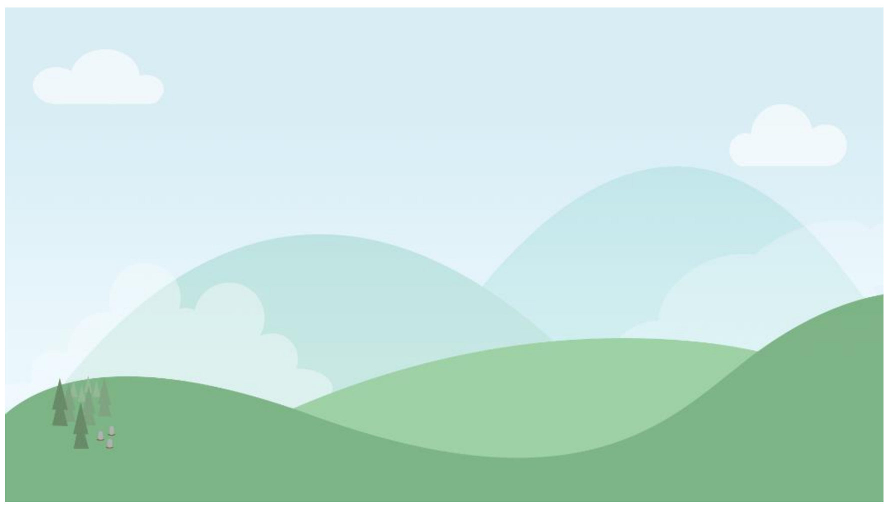


*Figure 3.5 - Menu background vector*

Figure 3.6 shows the Main Menu screen, as well as the Pause Menu overlaid upon the game. The overall window has been tinted darker to provide more contrast to make the Pause Menu options clearer. The menu also features custom fonts for the menu title (Buhry, 2006) and the button text (Murphy, 2011), with the menu titles in orange to match with the buttons.

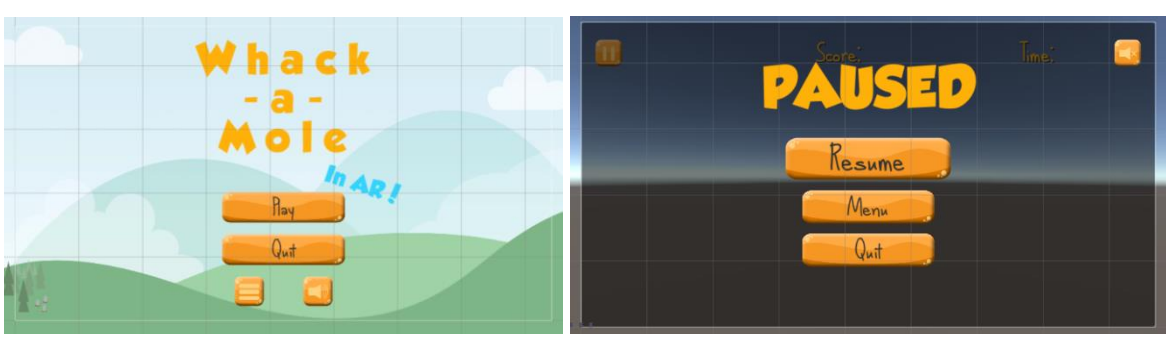



*Figure 3.6 – Menu screens in the editor*

To allow the Mute button to change between icons, it is created as a 'toggle' rather than a standard 'button'. The button directly controls the 'mute' Boolean operator in the Audio Source parented to the main camera.

### 3.1.3 Core Game

Due to the nature of the project, the focus was on the behaviour of the moles to create a playable game, as well as creating it within an augmented reality environment. As such the game will have no 3D modelled assets; however, these could easily be implemented at a later stage if this game were to be undertaken by a full development team.

#### *3.1.3.1 Mole*

The final asset for the mole includes the mole itself and the hole it comes out and retreats to. Collectively, this will be the 'Mole Unit', and making use of Unity's prefab asset type, the Mole Unit GameObject can be stored and instantiated through code. The C# script 'Mole_Script' is a component of the Mole Unit and contains routines for the movement and whether or not the mole has been hit.

##### 3.1.3.1.1 Pseudocode

The below pseudocode dictates the core functions that are used for the mole. Additional code will be required to tailor it for the Unity game engine, such as using the inbuilt 'Start()' and 'Update()' functions.

```
Func MoleBehaviour()
{
    MoveUp()
    WaitForHit()
    MoveDown()
}

Func MoveUp()
{
    while mole is in hole {
        move mole up
    if mole is at maximum height
        stop
    }
}

Func MoveDown()
{
    while mole is above hole {
        move mole down
    if mole is below hole
        stop
```

```
    }
}

Func WaitForHit()
{
    delayTimer = 0
    while moleHit == false and delayTimer<timeLimit {
        Increment delayTimer
    }
}
```

Code 3.1 - Pseudocode for Mole

#### 3.1.3.2 Game Manager

The game itself is managed by an overarching 'GameManager_Script' that contains the main game loop which oversees functionality such as the time limit of the game, when the moles appear, and how many moles can be 'active' at any one time.

##### 3.1.3.2.1 Pseudocode

```
Func MainGameLoop()
{
    while not game over {
        If timeLeft <= 0
            Game over
        Wait a random amount of time between 1 and maxTime
        If(CanActivateMoles()) {
            Loop moles
                If mole not active
                    Increment availableMoles
            End Loop
            If availableMoles > 0 {
                Pick random mole
                If random mole is active
                Pick new random mole
                Activate mole with random wait time
            }
        }
    }
}

Func CanActivateMoles()
{
    Loop moles
        If mole is active
            Increment activeMoles
    End Loop
    If activeMoles < maxActive
        Return true
    Else
        Return false
}
```

Code 3.2 - Pseudocode for GameManager

### 3.1.4 Augmented Reality

As discussed in Chapter 2, the augmented reality development platform Vuforia is integrated into Unity 5, making the development process faster and more streamlined. Vuforia allows for the implementation of augmented reality into the game to be a quick process where the core actions required are done through the developer portal on the website. Using Brovision's (2013) Augmented Reality Marker Generator to create a unique image target to upload, Vuforia then converts it into a Unity package containing all the database files required to use it. The actual implementation of augmented reality is a drag-and-drop process to set up the scene.

For a player to be able to interact with the game requires ray tracing from the camera to the target. "*Raycasting is the process of shooting an invisible ray from a point, in a specified direction to detect whether any colliders lay in the path of the ray*" (Raycasting Beginner Physics, n.d.). As Unity is a game engine, the process of creating the ray tracing script is with the 'Ray' and 'RaycastHit' classes, as well as 'Physics.Raycast' to detect the ray hit location (Unity - Scripting API: Physics.Raycast, 2017). The camera script therefore will create a ray with the mouse pointer or touch point as the hit location of the ray and evaluate if a mole was hit.

## 3.2 Evaluation of Design

It is entirely possible to test the product purely from a functional perspective i.e. if the menus work, if the behaviour of the mole is as expected, through black-box testing; to test the product in its entirety requires external feedback to evaluate it as a game, as outlined in Chapter 2. Both the inputs and expected outputs for black-box testing, as well as the questionnaire for participants are drafted below.

### 3.2.1 Black-Box Testing

| Input | Expected Outcome |
|---|---|
| Click menu buttons | Menus transition correctly |
| Point camera at target image | Game is displayed in AR |
| Click mole | Mole retreats and score increments |
| Timer reaches 0 | Game ends and final score is displayed |

*Table 3.1 - Input and outcome table*

The final data-input table for black-box testing the product will be more detailed and include more input lines to thoroughly test the expected functionality of the product.

### 3.2.2 Questionnaire

The following questions are rated on a scale of 1 - 5 where 1 is 'definitely disagree' and 5 is 'definitely agree' and focus on usability:

1. The menus were easy to use
2. The game loaded properly on the target image
3. The app was easy to use
4. The game responded quickly

The following questions are rated on a scale of 1 - 5 where 1 is 'definitely disagree' and 5 is 'definitely agree' and focus on your experience with the game. There is also a section below each one for additional comments.

1. The game was enjoyable
2. The augmented reality feature adds uniqueness to the game
3. The game difficulty was right

The above questionnaire is a rough guide to what the final questions shall be and will be created through external means such as OneDrive Forms or Survey Monkey as they offer in-built tools to analyse the results. A mix of quantitative and qualitative responses will better inform the final evaluation.

# 4 IMPLEMENTATION

## 4.1 INTRODUCTION

This chapter extends the design chapter, delving into further detail of the setup of the project, Unity-specific features, and the C# scripts used to control the game. The chapter focuses on the back-end of the game and how the game scripts interact with one-another, as well as how ray tracing turns it into a playable game. It also covers the front-end of the game and how the user interface displays time-changing variables such as the score. The code provided in this chapter is part of the final product of which can be found in the appendices.

## 4.2 PROJECT SETUP

### 4.2.1 Vuforia

To set up Vuforia, a developer account is required that will provide a key to use in Unity (Figure 4.1), as well as a database to store image targets. Within the website is the option to create the image target as shown in Figure 3.1 in Chapter 3, and it is that image target that Unity will then use to display the game upon. The last step to setting Vuforia up within Unity is to download and import the database file into the project.

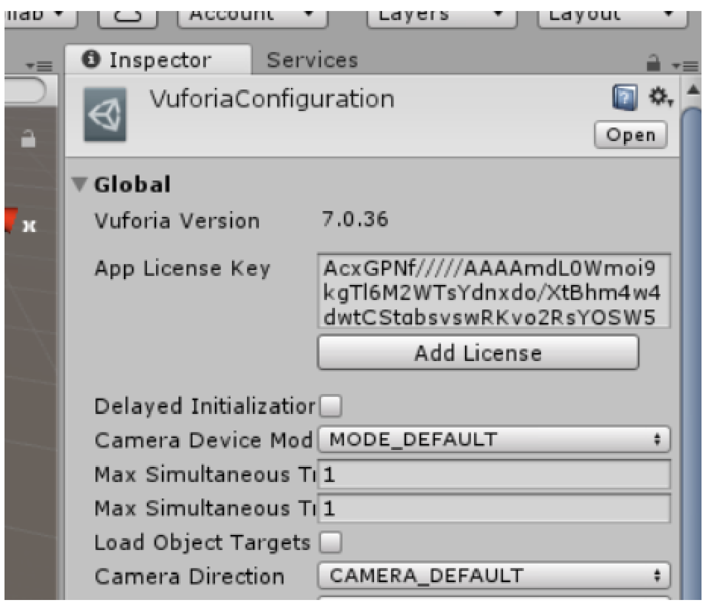



*Figure 4.1 - Importing Vuforia license key*

### 4.2.2 Unity

The default scene in Unity contains a main camera and a single directional light, which is unnecessary, as working with Vuforia the API creates an 'ARCamera' that has the Vuforia Behaviours component attached to it. Therefore, to set up Unity to work with the API, it is simply a case of installing it alongside the Unity Editor and activating it within the product due to Unity's inbuilt support for Vuforia, as mentioned in Chapter 2.

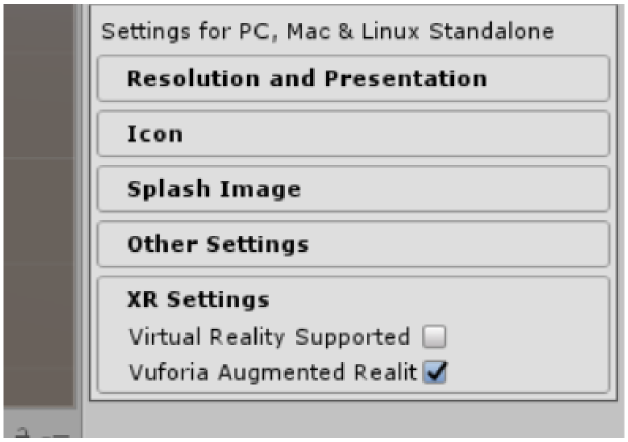



*Figure 4.2 - Vuforia enabled within Player Settings*

With Vuforia enabled and set up, the game can then be built as if it were not going to be in augmented reality. When testing the game on a computer, Unity will access the web-camera to simulate a mobile device's camera. To have the game display in augmented reality, an ImageTarget prefab needs to be placed into the scene, with the relevant image set in the inspector. Any 3D models to be rendered in augmented reality need to be a child of the ImageTarget prefab (Figure 4.3) and will then be displayed on the screen relative to their location from the ImageTarget prefab.

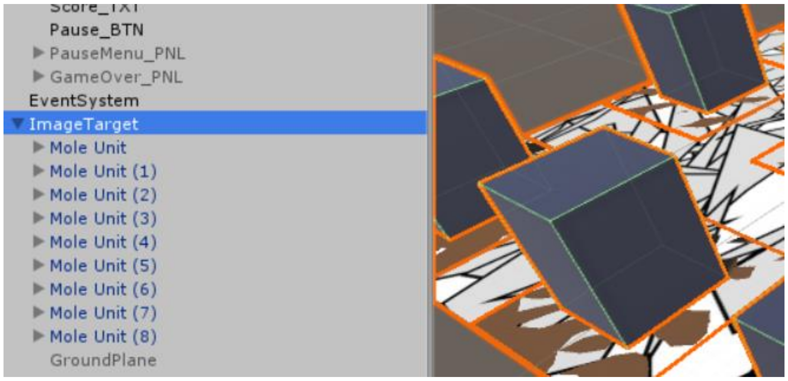



*Figure 4.3 - The hierarchy shows Mole Units as children of the ImageTarget*

## 4.3 USER INTERFACE

Unity's Graphic User Interface editor makes the UI creation quick and easy, placing the individual assets into the canvas with locations relative to one of the nine available anchor points. Some UI objects require a separate script to control the OnClick() actions, which are detailed below.

### 4.3.1 UI Assets

The smaller buttons such as the Credits and Pause images are individual assets (Figure 4.4). This contrasts with the primary buttons where there is a singular asset that is resized as needed (Figure 4.5), with the text changed within the editor.

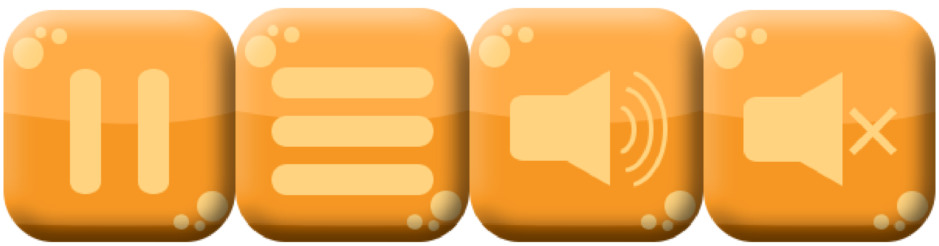


*Figure 4.4 - Pause, Credits, Sound On, Sound Off assets*

*Figure 4.5 - Main button asset*

All the assets once imported into the game engine are then set to the texture type 'sprite' to use within the GUI editor as shown in Figure 4.6.

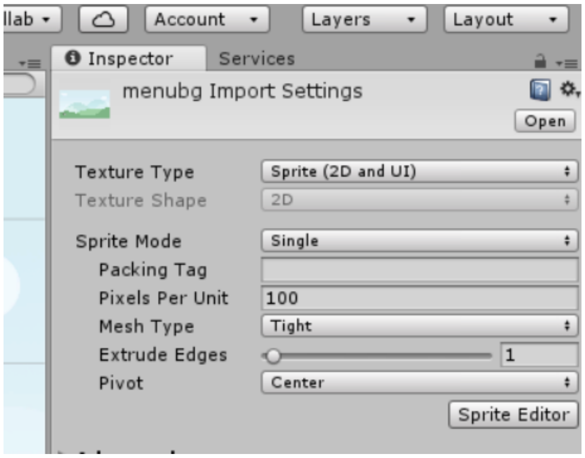



*Figure 4.6 - Texture type set to sprite*

### 4.3.2 Editor

#### *4.3.2.1 Main Menu*


*Figure 4.7 - Main menu*

The title text uses the same orange colour that features on the buttons, with the “In AR!” extra caption in a more saturated version of the blue used in the background image. Having the additional caption in blue breaks up the column of orange, while simultaneously drawing attention to the fact the game is in augmented reality.

This menu is a separate scene file in Unity, with the scene index set to zero and loading first in the build settings as shown in Figure 4.8.

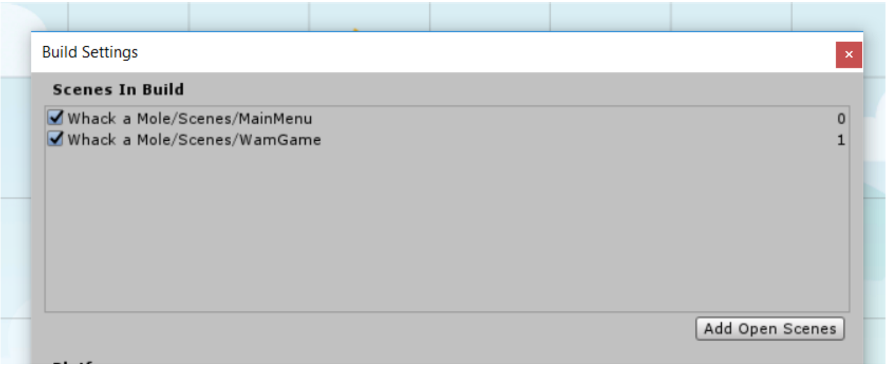



*Figure 4.8 - Project build settings*

#### 4.3.2.2 Pause Menu


Figure 4.9 - Pause menu in the editor

Using panels to separate the Pause Menu from the Game Over menu and the HUD allows the PauseMenu_Script to activate and deactivate the individual panel when it is called with the Pause button itself. The colour options for the panel are set to black with a high alpha value so the game itself is tinted darker.

#### 4.3.2.3 Game Over Menu

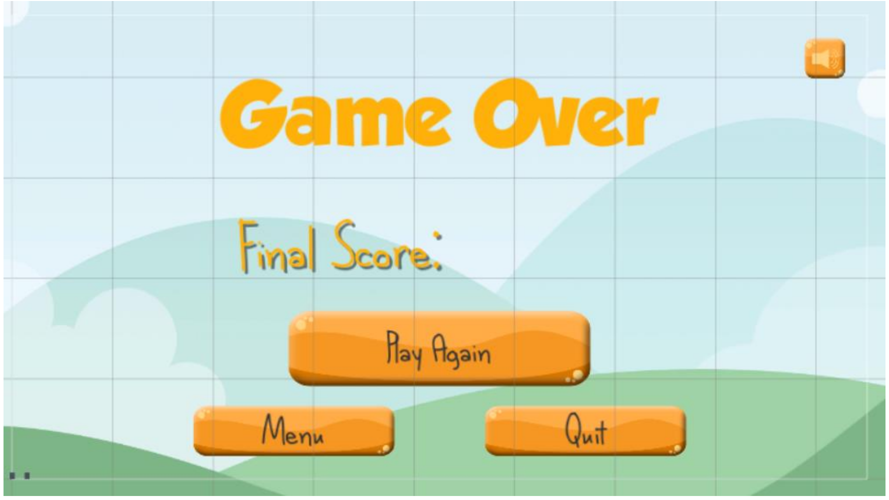



Figure 4.10 - Game Over menu

As the Score_Script is within the actual game scene itself, the Game Over menu is a panel within the game's user interface. The script for the buttons is the PauseMenu_Script to avoid rewriting the button functions unnecessarily.

#### *4.3.2.4 HUD*

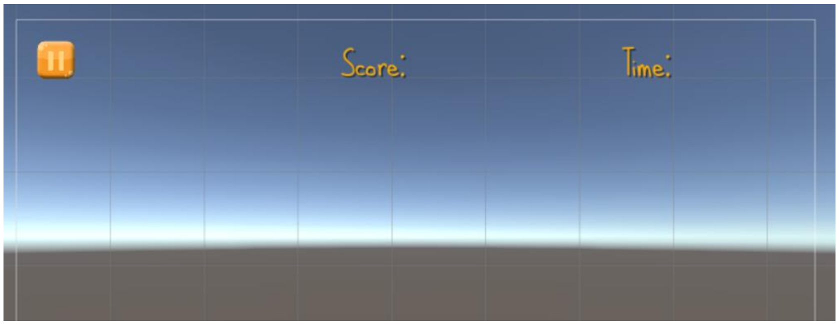



*Figure 4.11 - HUD in the editor*

Scripting for the Pause button is controlled by the PauseMenu_Script, while the score and time text objects have individual scripts controlling what is displayed.

### 4.3.3 Credits

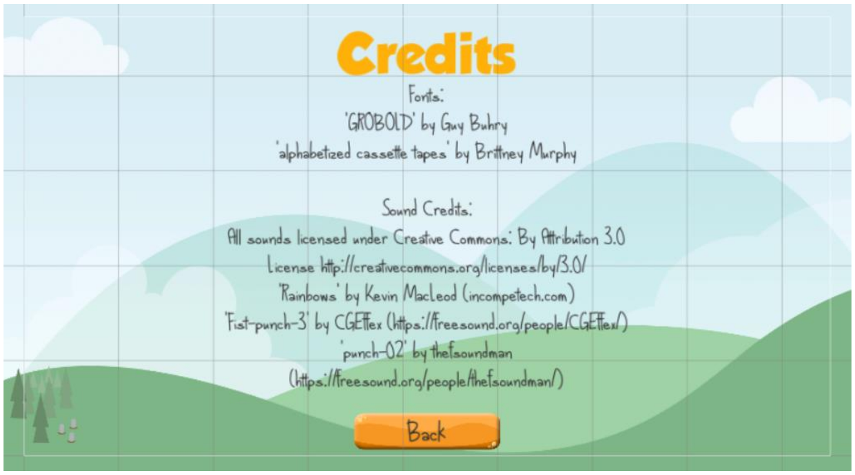



*Figure 4.12 - Credits screen accessed from the Main Menu*

The credits are part of the Main Menu scene and are kept within a separate panel within the menu canvas.

#### *4.3.3.1 Fonts*

The fonts used in the game are listed below:

- *GROBOLD* by G. Buhry, used for titles
- *alphabetized cassette tapes* by B. Murphy, used for all other text

#### *4.3.3.2 Sounds*

The sounds in the game are used under the Creative Commons Attribution (Creative Commons — Attribution 3.0 Unported — CC BY 3.0, n.d) license and are listed below:

- *Fist Punch 3* by CGEffex through freesounds.org
- *Rainbows* by K. MacLeod through incompetech.com
- *Punch 02* by thefsoundman though freesounds.org

### 4.3.4 Scripts

#### *4.3.4.1 Main Menu*

The MainMenu_Script takes in two GameObjects, 'MainMenuPanel' and 'CreditsPanel', which allow the different panels to be activated and deactivated with various button functions, and are both set in the editor itself. The Awake() function ensures that the CreditsPanel is hidden on start up. The remaining functions are called within the individual buttons' OnClick() function and navigate the menu accordingly. ShowCredits() and CloseCredits() navigate between the main menu and the credits panel, while StartGame() navigates to the "WamGame" scene which contains the game.

```
public class MainMenu_Script : MonoBehaviour {

    public GameObject CreditsPanel;
    public GameObject MainMenuPanel;

    void Awake()
    {
        MainMenuPanel.SetActive(true);
        CreditsPanel.SetActive(false);
    }

    public void StartGame()
    {
        SceneManager.LoadScene("WamGame");
    }

    public void QuitGame()
    {
        Application.Quit();
    }

    public void ShowCredits()
    {
        MainMenuPanel.SetActive(false);
        CreditsPanel.SetActive(true);
    }
```

```
        public void CloseCredits()
        {
            CreditsPanel.SetActive(false);
            MainMenuPanel.SetActive(true);
        }
    }
```

*Code 4.1 - MainMenu_Script*

#### *4.3.4.2 Pause Menu*

Similar to the MainMenu_Script, this script takes the Pause panel and Game Over panel set in the editor to control as and when each panel is active. The Update() function checks for the static Boolean from the GameManager_Script, and if true, will display the Game Over panel. Likewise, this script has a static Boolean for the pause status which is used across other scripts in the game.

```
void Start () {
    IsPaused = false;
     PauseMenu.SetActive(false);
     GameOverMenu.SetActive(false);
     MuteButton.SetActive(false);
}

private void Update()
{
     if (GameManager_Script.GameOver)
    {
          GameOverMenu.SetActive(true);
         MuteButton.SetActive(true);
    }
}

public void Pause()
{
     PauseMenu.SetActive(true);
    MuteButton.SetActive(true);
     IsPaused = true;
}

public void Resume()
{
     PauseMenu.SetActive(false);
    MuteButton.SetActive(true);
     IsPaused = false;
}

public void MainMenu()
{
     SceneManager.LoadScene("MainMenu");
}

public void Quit()
{
     Application.Quit();
}

public void RestartScene()
{
```

```
        SceneManager.LoadScene(SceneManager.GetActiveScene().name);
    }
```

*Code 4.2 - PauseMenu_Script*

#### *4.3.4.3 HUD*

Both the Score_Script and Timer_Script are components of the GameManager object and each script contains static variables 'score' and 'currentTime', respectively. The Update() functions of each script update the relevant text objects.

```
public class Score_Script : MonoBehaviour {

    private static int score;
    public Text scoreText;
    public Text finalScoreText;

    public static int Score
    {
        get
        {
            return score;
        }

        set
        {
            score = value;
        }
    }

    // Use this for initialization
    void Start () {
        score = 0;
      }

      // Update is called once per frame
      void Update () {
        scoreText.text = "Score: " + score.ToString();
        finalScoreText.text = "Final Score: " + score.ToString();
      }
}
```

*Code 4.3 - Score_Script*

The Timer_Script's update function contains a pause status check which will stop the timer from counting down if the game is paused.

```
if (!PauseMenu_Script.IsPaused)
            currentTime -= Time.deltaTime;
```

*Code 4.4 - If statement to check pause status*

It also has a check to set the currentTime value to zero if the countdown goes below it.

```
if (currentTime <= 0)
            currentTime = 0;
```

*Code 4.5 - If statement to check currentTime value*

## 4.4 Game Scripts

### 4.4.1 Main Game Loop

Run from the Update() function of the GameManager_Script, which is a component of the empty game object GameManager, the RunGame() function itself is only called provided the game is not over or paused.

```
if (!GameOver && !PauseMenu_Script.IsPaused)
{
    RunGame();

    // check if time up
    if (Timer_Script.CurrentTime <= 0)
        GameOver = true;
}
```

*Code 4.6 - If statement to determine whether the game should be run*

Since the function is called from within Update(), it cannot be of type IEnumerator to be run as a Unity Coroutine like the mole's behaviour functions can. As such, variables and Unity's time keeping are used to create a delay before checking and activating a new mole. CanActivateBool() is the function described in the pseudocode in Chapter 3 that determines if a mole can be activated or not based on how many are currently active. An 'if statement' checks the number of inactive moles and if greater than zero, randomly selects an inactive mole. To make the game more varied, the hit wait time is randomised between 0.5 and two seconds.

```
private void RunGame() // main game loop
{
    // wait a random time before activating new mole
    float delayTimer = 0f;
    float currentSpawnDelay = Random.Range(1, (maxSpawnDelaySec * 1000)) /
1000;
    while (delayTimer < currentSpawnDelay)
    {
        delayTimer += Time.deltaTime;
    }

    // check if there are moles to activate
    if (CanActivateBool())
    {
        // activate random inactive mole
        if (MolesInactive() > 0)
```

```
            {
                int random = (int)Random.Range(0, moleList.Count);
                while (moleList[random].ReturnMolePrefab().activeSelf) //
generate new random until selected mole is inactive
                    random = (int)Random.Range(0, moleList.Count);

                // randomise mole hit wait time
                float newHitTime = Random.Range(0.5f, 2);
                moleList[random].ActivateMole(newHitTime);
            }
        }
    }
```

*Code 4.7 - RunGame() function*

If a valid hit is detected from the ray tracing, covered later in this chapter, the function MoleHit (Mole_Script m) is called which handles adding score and calling the targeted mole's Whack() function.

```
private void MoleHit(Mole_Script m)
{
    Score_Script.Score += m.Whacked() ? 0 : 10; // only add score if mole
isnt "whacked"
    m.Whack();
}
```

*Code 4.8 - MoleHit (Mole_Script m) function*

### 4.4.2 Mole Script

The Start() function of the mole initialises the location of the mole to set it below the hole and deactivates it, ready to be activated from the Game Manager. It also calls the Game Manager's AddMole function to "register" itself. This allows flexibility in the number of moles in the game itself using lists that are described above.

```
void Start () {
    MolePrefab.GetChild(0).gameObject.SetActive(false);
    timeLimit = 1.0f;
    speed = 30.0f;

    // set up and down yPos values
    float height = MolePrefab.GetComponent<Collider>().bounds.size.y; //
get bounding box size for mole
    upPos = height / 2;
    downPos = -(height / 2);

    // move mole below hole
    Vector3 temp = MolePrefab.localPosition;
    temp.y = downPos;
    MolePrefab.localPosition = temp;
```

```
        MolePrefab.gameObject.SetActive(false);
        GameManager_Script.Instance.AddMole(gameObject); // allows for flexible
number of moles
    }
```

*Code 4.9 - Start() function of a mole*

The behaviour of an individual mole is controlled through three different coroutines activated in succession by MainMoleLoop() if the mole itself is made active. These coroutines move the mole up, wait for it to be hit or for the time limit to be up, and followed by moving it back down. Given the nature of coroutines, within the 'while loops' of each is a check for the pause status of the game to prevent any actions carried out while the game is paused.

```
    private IEnumerator MainMoleLoop() // core mole actions
    {
        yield return StartCoroutine(MoveMoleUp());
        yield return StartCoroutine(WaitForHit());
        yield return StartCoroutine(MoveMoleDown());
    }

    private IEnumerator MoveMoleUp()
    {
        while (MolePrefab.localPosition.y < upPos) // while mole is below hole
        {
            if (!PauseMenu_Script.IsPaused)
            {
                Vector3 tempPos = MolePrefab.localPosition;
                float newYPos = tempPos.y + (speed * Time.deltaTime); // update
y pos based on speed
                tempPos.y = newYPos > upPos ? upPos : newYPos; // if newYPos >
height, set tempPos.y to height
                MolePrefab.localPosition = tempPos;
            }

            yield return null;
        }
    }

    private IEnumerator WaitForHit() // mole remains static until time is up or
hit
    {
        float time = 0.0f;
        while (!whacked && time < timeLimit) // loop while mole not hit and
time not up
        {
            if (!PauseMenu_Script.IsPaused)
            {
                time += Time.deltaTime;
            }
            yield return null;
        }
    }

    private IEnumerator MoveMoleDown()
    {
        while (MolePrefab.localPosition.y > downPos) // while mole is above
```

```
    {
        if (!PauseMenu_Script.IsPaused)
        {
            Vector3 tempPos = MolePrefab.localPosition;
            float newYPos = tempPos.y - (speed * 2 * Time.deltaTime); //
update y pos based on speed
            tempPos.y = newYPos < downPos ? downPos : newYPos; // if
newYPos < 0, set tempPos.y to 0
            MolePrefab.localPosition = tempPos;
        }

        yield return null;
    }

    MolePrefab.gameObject.SetActive(false); // deactivate once back in hole
}
```

*Code 4.10 - Mole coroutines*

If a valid hit is detected by the Game Manager, the Whack() function is called from it which then sets the 'whacked' variable to true. It also activates a small text pop up (Figure 4.13), and plays a sound effect to provide feedback to the player.

```
public void Whack()
{
    whacked = true;
    MolePrefab.GetChild(0).gameObject.SetActive(true);
    GetComponent<SFX_Script>().PlayMoleHit();
}
```

*Code 4.11 - Whack() function*

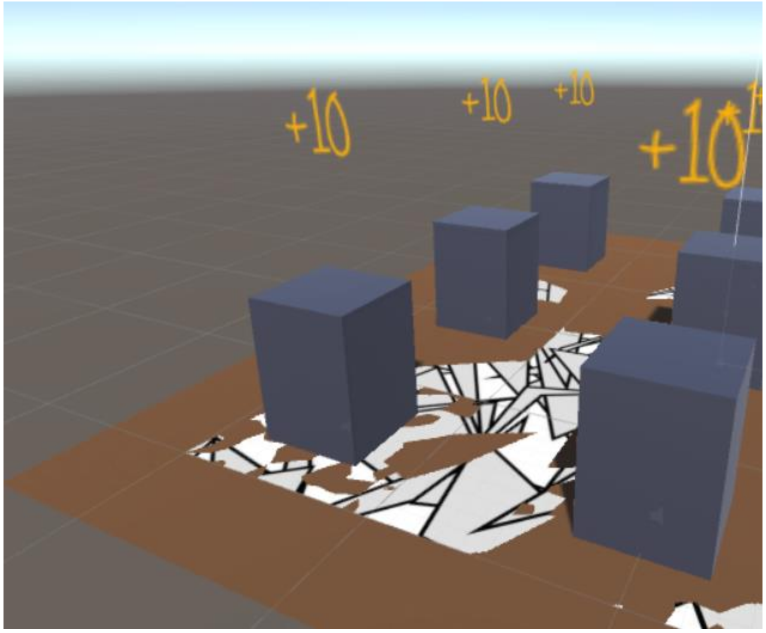



*Figure 4.13 - Text pop up on mole*

### 4.4.3 Ray Tracing

As mentioned in the previous chapter, Unity's inbuilt raycasting related classes make hit detection a simple task. For testing purposes, there are two instances of the ray tracing script, one for mouse-clicks and one for touches. The code below illustrates the ray tracing and the alternate lines that change it to work for touch-detection.

```
if (Input.GetMouseButtonDown(0))
{
    RaycastHit hit = new RaycastHit();
    Ray ray = arGameCam.ScreenPointToRay(Input.mousePosition);
    if (Physics.Raycast(ray, out hit))
```

*Code 4.12 - Ray tracing for mouse*

```
if (Input.touchCount > 0)
{
    RaycastHit hit = new RaycastHit();
    foreach (Touch t in Input.touches)
    {
        Ray ray = arGameCam.ScreenPointToRay(t.position);
        if (Physics.Raycast(ray, out hit))
```

*Code 4.13 - Ray tracing for touchscreens*

The Ray 'ray' takes the screen position of the mouse or touch point, relative to the camera, and converts that into a ray to use in raycasting. The Raycast Out variable 'hit' stores the information of the selected object, if any. Since the game needs to know if the hit object is a mole, only the GameObject components are compared; if there is a match, the score is incremented by ten, and the mole is 'whacked'.

```
foreach (Mole_Script m in moleList)
{
    if (m.ReturnMolePrefab().activeSelf && m.ReturnMolePrefab() ==
hit.transform.gameObject)
    {
        MoleHit(m);
    }
}
```

*Code 4.14 - If statement checks the hit object is a mole*

## 4.5 AUDIO

### 4.5.1 Background Music

The main cameras in both the Main Menu and the Game scenes have an empty GameObject 'MusicPlayer' parented to them containing an audio source (Figure 4.14). These audio sources have the background music *Rainbows* set as the default audio clip and are set to both Loop and Play On

Awake. Play on Awake ensures the music plays on starting and Loop will prevent the music ending leaving the remainder of the game silent.

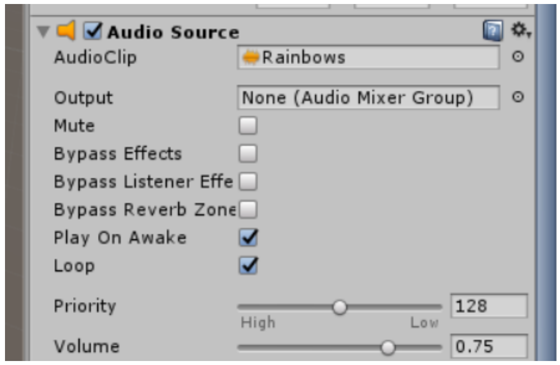



*Figure 4.14 - Audio source settings in the Inspector*

### 4.5.2 Sound Effects

A default audio source and the SFX_Script components, described below, are attached to each mole (Figure 4.15). These components are used to play the sound effects when required, called from the main Mole_Script when it is hit.

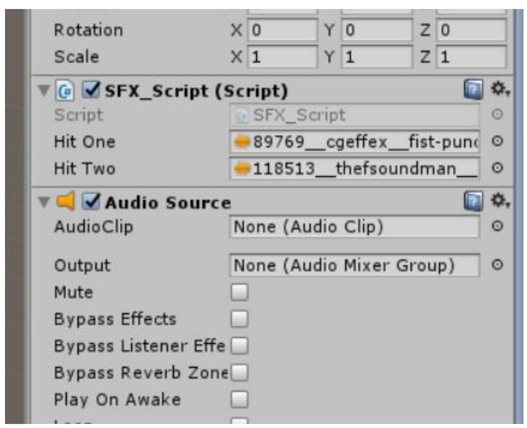



*Figure 4.15 - Sound effects components on a mole*

#### *4.5.2.1 SFX Script*

Taking two audio clips set in the editor, this script sets 'source' to the audio source component on the GameObject in Awake(); the function PlayMoleHit() randomises the volume, pitch, and which of the two audio clips to play when called.

```
public class SFX_Script : MonoBehaviour {

    public AudioClip hitOne;
    public AudioClip hitTwo;

    private AudioSource source;
```

```
    private float lowPitch = 0.75f;
    private float highPitch = 1.25f;
    private float lowVol = 0.5f;
    private float highVol = 1.0f;

      // Use this for initialization
      void Awake () {
        source = GetComponent<AudioSource>();
      }

    public void PlayMoleHit()
    {
        // randomise hit sound
        source.pitch = Random.Range(lowPitch, highPitch);
        source.volume = Random.Range(lowVol, highVol);
        if (Random.Range(1, 50) / 2 == 2)
            source.PlayOneShot(hitOne);
        else
            source.PlayOneShot(hitTwo);
    }
}
```

*Code 4.15 - SFX_Script*

# 5 EVALUATION

> *Software testing is performed to verify that the completed software package functions according to the expectations defined by the requirements/specifications. The overall objective to not to find every software bug that exists, but to uncover situations that could negatively impact the customer, usability and/or maintainability.*

(Software Testing - Its Purpose and Definition, n.d.)

## 5.1 BLACK BOX TESTING

"*Black-box testing of modules refers to the unit testing that is performed using requirement-based tests to uncover the program errors in external functions, behaviours, and interfaces*." (Gao et al., 2003) This method of testing with the game will go through a user's expected interactions with the application. The following table shows the test, expected results, and actual results carried out on the game.

| Test Name | Execution | Expectation | Result | Comments |
|---|---|---|---|---|
| **Start** | Run the program | The program will start, and the Main Menu will be displayed | As expected | |
| **Show Credits** | Press the Credits button from the Main Menu | The Main Menu will be hidden, and the Credits screen will be displayed | As expected | |
| **Close Credits** | Press the Back button from the Credits screen | The Credits screen will be hidden, and the Main Menu will be displayed | As expected | |

| | | | | |
|---|---|---|---|---|
| **Mute** | Press the Mute button from the Main Menu while the audio is unmuted | The audio will be muted but continue to play | As expected | Does not remain muted when transitioning to the Game |
| **Unmute** | Press the Mute button from the Main Menu while the audio is muted | The audio is unmuted at continues to play as normal | As expected | Does not remain unmuted when transitioning to the Game |
| **Play Game** | Press the Play button from the Main Menu | Main Menu closes, and the screen displays the camera feed | As expected | |
| **Quit** | Press the Quit button from the Main Menu | The program will exit | As expected | |
| **Display Game** | Point camera at the ImageTarget | 3D models for the game will be displayed on top of the Image Target | As expected | |
| **Pause Game** | Press the Pause button while the game is unpaused | The game will be paused including animations and the countdown timer | As expected | |
| **Mute** | Press the Mute button from the Pause | The audio will be muted but continue to play | As expected, though does not | Does not remain muted when |

| | | | | |
|---|---|---|---|---|
| | Menu while the audio is unmuted | | mute sound effects | transitioning to the Main Menu |
| **Unmute** | Press the Mute button from the Pause Menu while the audio is muted | The audio is unmuted at continues to play as normal | As expected | Does not remain unmuted when transitioning to the Main Menu |
| **Resume Game** | Press the Resume button from the Pause Menu while the game is paused | Game will resume from its current point, including the animations and countdown timer | As expected | |
| **Return to Menu** | Press the Menu button from the Pause Menu | Game closes and the Main Menu is loaded and displayed | As expected | |
| **Quit** | Press the Quit button from the Pause Menu | The program will exit | As expected | |
| **Successfully hit mole** | Hit a mole once it has fully risen | Orange "+10" text will display, score will increment by 10, and mole will withdraw | As expected, though "+10" may display backwards depending on the orientation of the image target | |

| | | | | |
|---|---|---|---|---|
| **Successfully hit the same mole twice** | Hit a mole two consecutive times before it withdraws | Score only increments once no matter how many times it is hit afterwards | As expected | |
| **Unsuccessfully hit mole** | Hit any location on the screen that is not a mole | Nothing happens | As expected | |
| **Game Over** | Wait for the countdown timer to get to zero | Game stops and Game Over screen is displayed, showing the final score | As expected | |
| **Play Again** | Press the Play Again button from the Game Over menu | Game scene is reloaded, and camera feed is displayed | As expected | |
| **Return to Menu** | Press the Menu button from the Game Over menu | Game Over screen closes and the Main Menu is loaded and displayed | As expected | |
| **Quit** | Press the Quit button from the Game Over menu | The program will exit | As expected | |

*Table 5.1 - Black box interaction tests*

## 5.2 USER SURVEY

The initial questionnaire from Chapter 3 has been reworded to fit the style and question options provided by Microsoft Forms. Additional questions have been added to better understand the users, as their backgrounds may influence how they view the game.

### 5.2.1 Questionnaire

1. How often do you play video games?
2. What do you play video games on?
3. How often do you play mobile games?
4. Have you heard of Augmented Reality before?
5. How experienced would you say you are with Augmented Reality?
6. On a scale of 1-5, how easy were the menus to use?
7. On a scale of 1-5, how clear were the menu buttons against the backgrounds?
8. Any additional thoughts about the menus?
9. On a scale of 1-5, how easy were the controls for the game?
10. Any additional thoughts about the controls?
11. Would you agree that the game displayed properly on top of the 'Image Target' properly?
12. Would you agree that the game loaded quickly?
13. On a scale of 1-5, how easy was the app to use in general?
14. Would you agree that Augmented Reality adds uniqueness to the game?
15. Would you agree that the game is enjoyable?
16. Is this a game you would recommend to other people?
17. How would you rate the difficulty of the game?
18. Any additional comments?

### 5.2.2 Hypotheses

- There will be few, if any, participants who will have not heard of augmented reality in one way or another
- Most participants will be inexperienced with augmented reality, as not many readily available applications outside of mobile devices make use of it
- Because of the colour scheme used for the user interface of the game, the consensus will be that the menu is clear and easy to use
- Most participants will agree that the augmented reality features of the game are worth having due to adding a uniqueness to the game itself

### 5.2.3 Results

The initial questions showed that the participants were of a varied group, as gauged from how often they play video games (Figure 5.1) and how often they play mobile games (Figure 5.2).

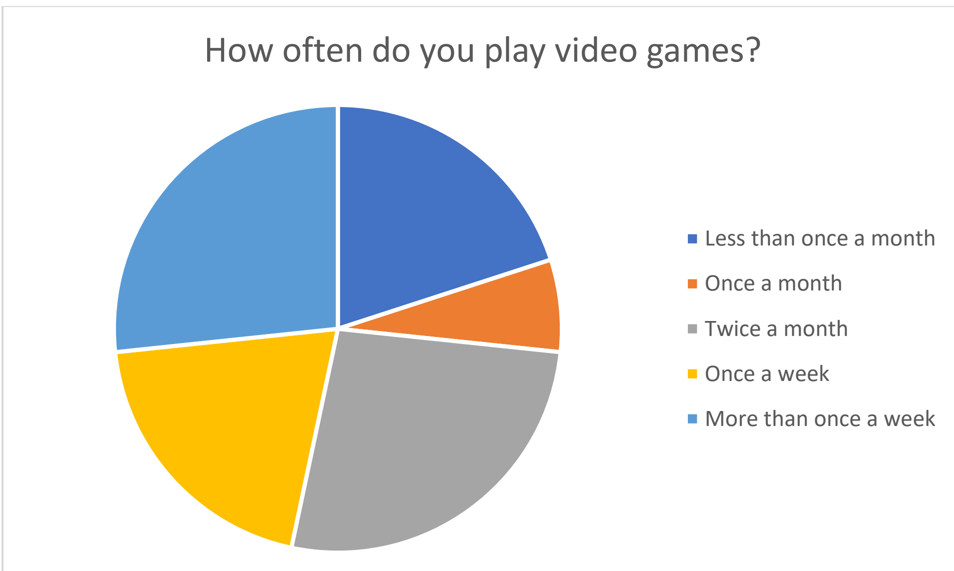



*Figure 5.1 - Participants' video game habits results*

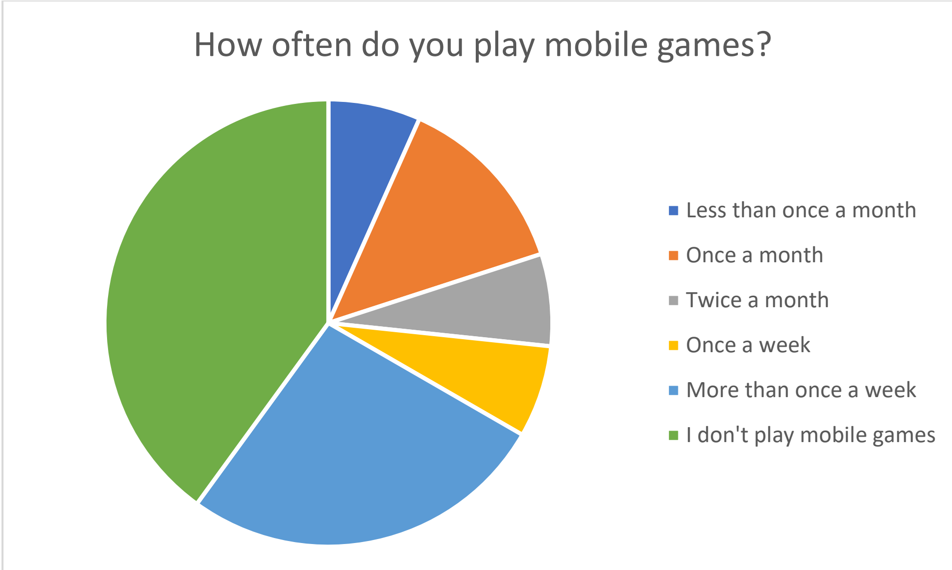



*Figure 5.2 - Mobile gaming habits results*

It can be seen in Figure 5.3 that the participants do not stick to just one device, with many having a console alternative alongside a mobile device. The variety of people allows for more general conclusions to be made about the quality of the game, and their comments about the project can be used to influence changes in any further releases.

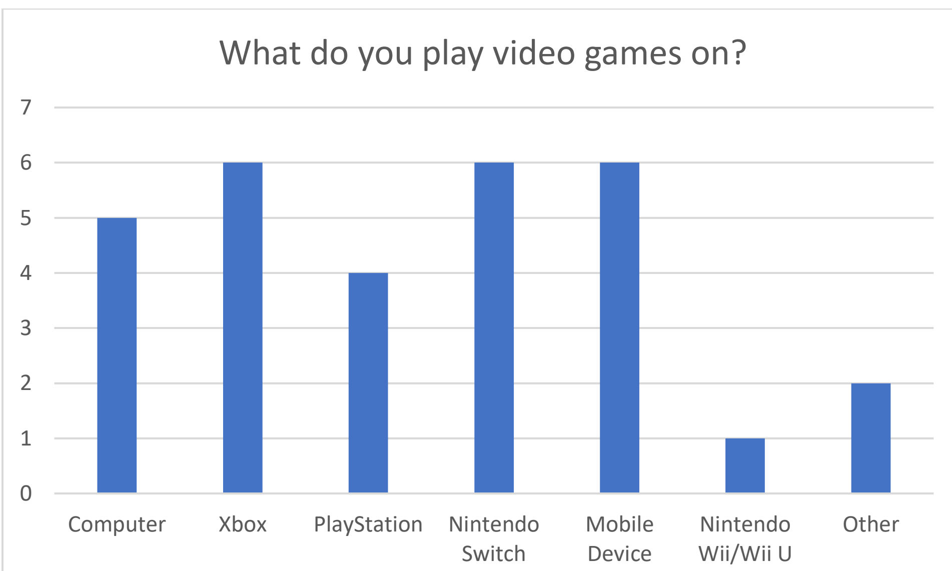



*Figure 5.3 - Video game devices used results*

As well as understanding the participants gaming habits, the questionnaire included examining their experience with augmented reality, with nearly all participants having heard of it (Figure 5.4) but not having much experience with it as shown in the results of the follow-up question (Figure

5.5). These results match the hypotheses about augmented reality described earlier, as it is a technology that is becoming more publicly available, as well as a key selling point for the recent mobile operating system update from Apple (Unreal Engine, 2017, Apple Inc, 2017).

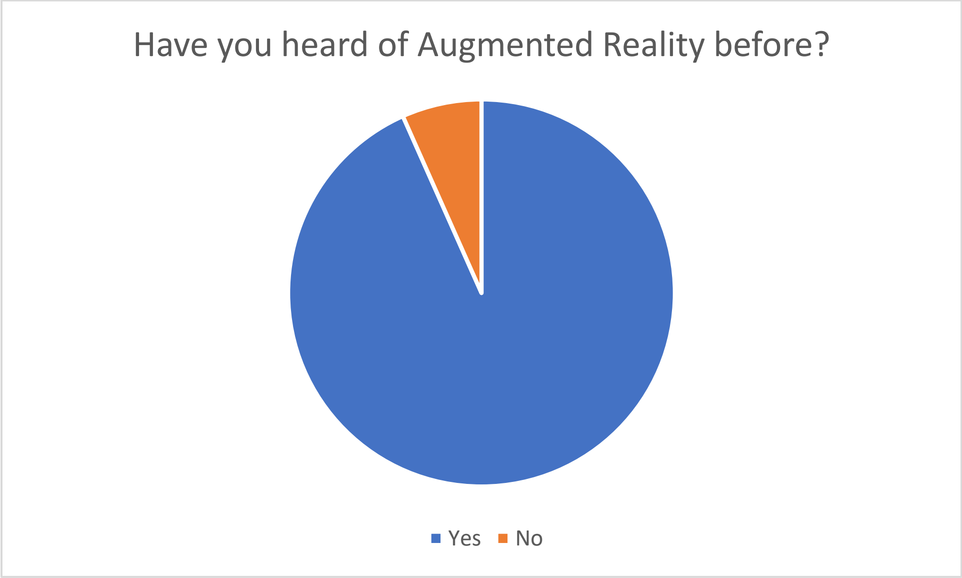



*Figure 5.4 - Knowledge of augmented reality results*

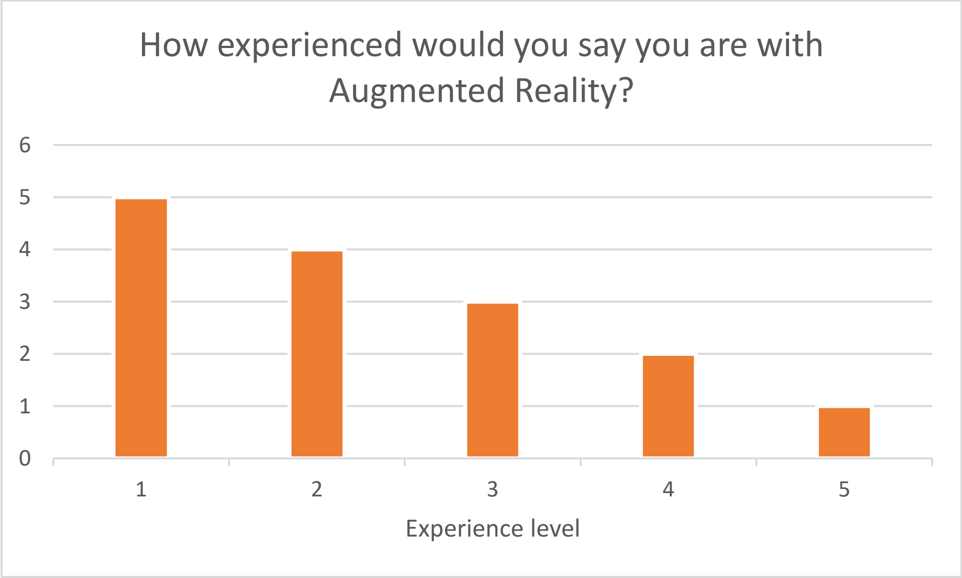



*Figure 5.5 - Experience with augmented reality results*

Using a 1 to 5 rating system, where one represented ‘not at all’ and five meant ‘very’, questions about the usability of the menu could be quantified. Figure 5.6 shows the results about the ease of use for the menu, button clarity of the menu, ease of the controls, and overall app usability. The results for the menu, alongside participant comments prove the hypothesis that the contrast between the orange buttons versus a pale blue background made the menu easy to use.

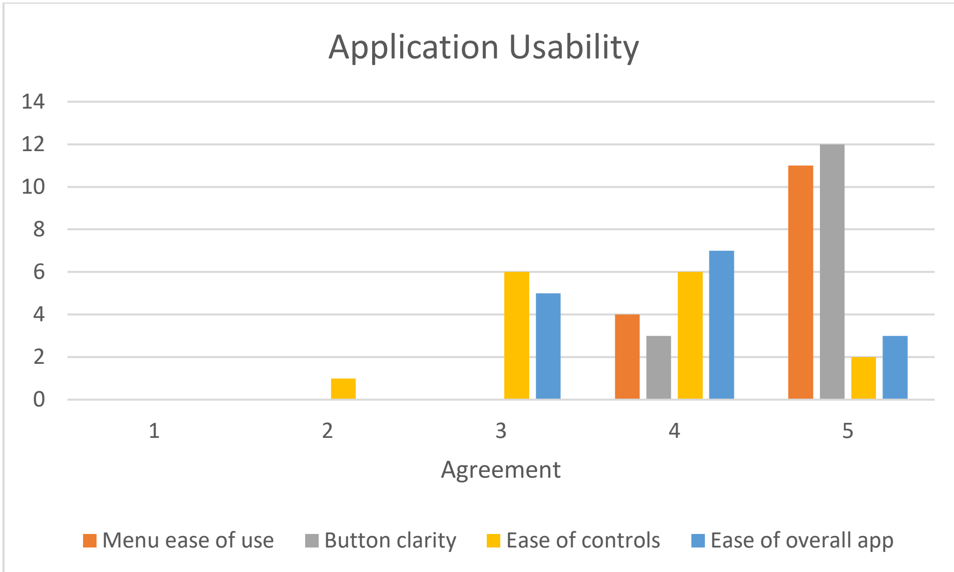



*Figure 5.6 - Application usability results*

Questions about usability included open-ended 'Any additional thoughts about…' to which eight participants provided additional feedback on the menu, and nine had feedback about the controls. Most menu comments reaffirmed the contrast and ease of use. Four of the eight comments, however, included that the font size on the credits screen was difficult to read due to the sizing, which would be something to modify in later versions. Comments about the controls were more varied, with the most frequent critique stating that the game was "unresponsive" and "it did not often feel like [the participant] was hitting the moles", which could be linked to the ray tracing used for the augmented reality interaction.

Using the same 1 to 5 scale as the previous questions, Figure 5.7 illustrates participants' opinion of the game, including the augmented reality features, the overall enjoyability of the game, as well as their likelihood to recommend the game to another person. Included in Figure 5.7 is the difficulty opinion, wherein 1 represents 'too easy' and 5 'too hard'. The hypothesis that participants will feel that the game is made unique by the augmented reality is proven by the overall positive responses received for the question. Over half the participants also positively responded to recommending the game to another, which suggests that with further improvements, all participants would respond with five if asked again.

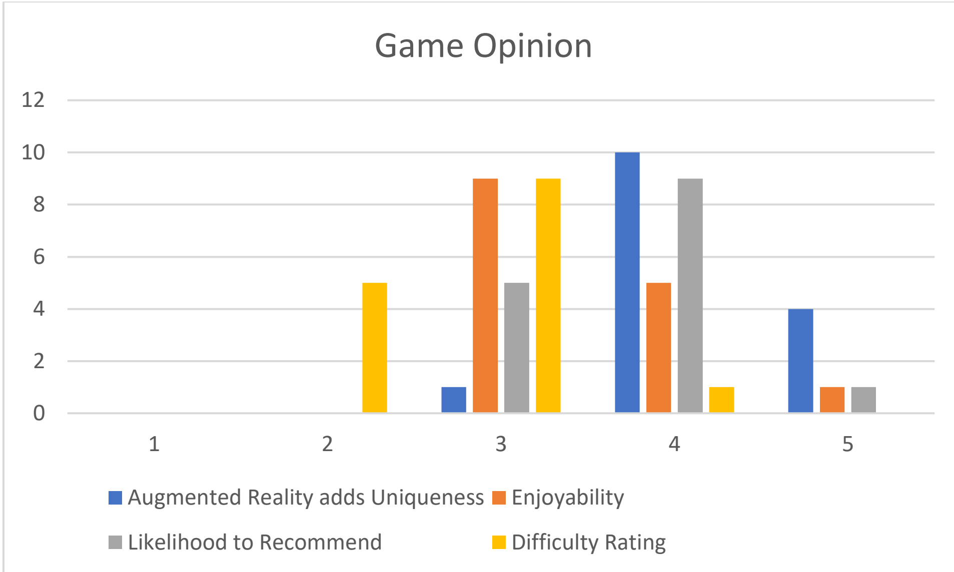



*Figure 5.7 - Game opinion results*

With over half of the participants marking the game at the average value of 3 could have been negatively affected by the control issues highlighted earlier. However, the average value of 3.47 out of 5 suggests that the game is enjoyable and with further improvements, could achieve a higher rating. While the average difficulty rating was 2.73 out of 5, a third of participants responded that the game was 'easy' as opposed to the value of 3 for 'just right', the comments from the following open-ended question provides insight into ways to improve the game to increase the number of responses that the game difficulty is 'correct'.

### 5.2.4 Conclusions

The questionnaire results confirm the hypotheses from earlier in the chapter, confirming that the project overall was successful. The participants' comments include ways to further improve the project for later revisions, including changes to UI elements, as well as additional features such as a high score board, different methods of scoring or penalising score, and a way to play the game in augmented reality without the image target.

# 6 CONCLUSION

## 6.1 LITERATURE REVIEW CONCLUSION

The purpose of the literature review was to evaluate concepts, existing products, and methods related to the project that would provide insight into what would be required to proceed with the project. In the end, the review proved highly successful in identifying the optimal game engine and augmented reality API for developing the project, as well as providing insight into augmented reality itself and developing a better understanding and appreciation for the technology.

## 6.2 DESIGN CONCLUSION

The design chapter provided reasoning and insight to the choices made for the interface of the project as well as pseudocode for major behaviours of the game. It also went further in-depth into augmented reality specifically for the Unity game engine and Vuforia focusing on a user being able to tap on various objects to interact with them, proving useful come the implementation stage.

## 6.3 IMPLEMENTATION CONCLUSION

Implementation of the project through Unity was achieved smoothly, with features not fully covered in the design chapter that came about through the specifics of the game engine itself. Due to budget restraints, the game has been created with standard 3D cubes as opposed to an imported 3D model with its own animations, such as one created in Autodesk Maya. A more thorough understanding of file saving and settings saving procedures in Unity and for smartphones would have allowed for high scores to be saved, and the audio mute setting to transfer between sessions. More experience with Unity for features such as online play, or the online sharing of scores would add replayability to the game through the form of creating rivalries and score targets. As the related functionality is written in separate C# scripts, implementing this additional functionality in a later revision should not pose a risk to the core game scripts.

## 6.4 EVALUATION CONCLUSION

User feedback from the questionnaire highlighted issues that had not been considered or found while performing the black box tests, such as when some users found that the hit locations did not always register properly. The feedback also identified issues that could not be recreated within the Unity developer, such as two of the moles not rising to the top as they should. In general, the User Interface performed as expected, being easy to use, especially on a small-screened device like a smartphone; with many comments about adding a 'High Scores' screen, which is something to consider in later revisions of the project. Overall the project was successful as shown from player feedback.

# 7 REFERENCES

9. The advantages and disadvantages of questionnaires. (n.d.) Libweb.surrey.ac.uk. [Online] [Accessed on 23 November 2017] http://libweb.surrey.ac.uk/library/skills/Introduction%20to%20Research%20and%20Managing%20Information%20Leicester/page_51.htm.

Advantages and Disadvantages of using Questionnaires. (2012) Compass.port.ac.uk. [Online] [Accessed on 23 November 2017] http://compass.port.ac.uk/UoP/file/fa9fbb2f-06fb-4fef-9ce1-c5e06b26a831/1/Questionnaires_IMSLRN.zip/page_07.htm.

Apple Inc (2017) *Introducing ARKit: Augmented Reality for iOS*. [video] [Accessed on 29 October 2017] https://developer.apple.com/videos/play/wwdc2017/602/.

App Showcase. (2017) Vuforia.com. [Online] [Accessed on 29 October 2017] https://vuforia.com/apps.html.

ARCore Overview. (2017) Google Developers. [Online] [Accessed on 29 October 2017] https://developers.google.com/ar/discover/.

ARKit - Apple Developer. (2017) Developer.apple.com. [Online] [Accessed on 29 October 2017] https://developer.apple.com/arkit/.

AR Features. (2017) Vuforia.com. [Online] [Accessed on 29 October 2017] https://vuforia.com/features.

Augmented Reality Marker Generator - Brosvision. (2013) Brosvision.com. [Online] [Accessed on 15 February 2018] http://www.brosvision.com/ar-marker-generator/.

BBC News - Bankers 'whacked' in arcade game. (2009) News.bbc.co.uk. [Online] [Accessed on 11 November 2017] http://news.bbc.co.uk/1/hi/england/suffolk/8410453.stm.

Bethesda Softworks LLC (2017) *Skyrim VR*. [image] [Accessed on 8 October 2017] https://media.playstation.com/is/image/SCEA/skyrim-vr-screenshot-02-us-12jun17?$MediaCarousel_Original$.

Bhat, A. (2016) Mobile gaming share to overtake PC gaming in 2016 with $36.9 billion revenue: Report. International Business Times, India Edition. [Online] [Accessed on 16 April 2018] https://www.ibtimes.co.in/mobile-gaming-share-overtake-pc-gaming-2016-36-9-billion-revenue-report-675763.

Black Box Testing - Software Testing Fundamentals. (2018) Software Testing Fundamentals. [Online] [Accessed on 14 February 2018] http://softwaretestingfundamentals.com/black-box-testing/.

Blueprints Visual Scripting. (n.d.) Docs.unrealengine.com. [Online] [Accessed on 25 October 2017] https://docs.unrealengine.com/latest/INT/Engine/Blueprints/.

Brewster, K. (2016) *Pokémon Go review: not a good game... but a great experience*. The Guardian. [Online] [Accessed on 10 November 2017] https://www.theguardian.com/technology/2016/jul/12/pokemon-go-review-it-may-not-be-a-good-game-but-its-a-great-experience.

Buhry, G. (2006) *GROBOLD*. [True Type Font].

CGEffex. (2010) *Fist Punch 3*. [Online] [Accessed on 6 April 2018] https://freesound.org/people/CGEffex/sounds/89769/download/89769__cgeffex__fist-punch-3.mp3.

Choo, J. (2016) *What are the main pros and cons of Unreal Engine 4 in general?*. Quora. [Online] [Accessed on 25 October 2017] https://www.quora.com/What-are-the-main-pros-and-cons-of-Unreal-Engine-4-in-general/.

Cowley, D. (2017) *High-End Augmented Reality Is Within Reach*. Unrealengine.com. [Online] [Accessed on 22 October 2017] https://www.unrealengine.com/en-US/blog/high-end-augmented-reality-is-within-reach.

Craig, A. (2013) *Understanding augmented reality*. 1st ed. Amsterdam: Elsevier, pp.1-35, 39-40.

Creative Commons — Attribution 3.0 Unported — CC BY 3.0. (n.d.) [Online] [Accessed on 5 April 2018] https://creativecommons.org/licenses/by/3.0/.

Crews, E. (2017) The Rise of Mobile Gaming. Geek Insider. [Online] [Accessed on 16 April 2018] http://www.geekinsider.com/rise-mobile-gaming/.

Denso Wave Incorporated (2013) *A QR Code symbol of this size can encode 300 alphanumeric characters.*. [image] [Accessed on 8 November 2017] https://web.archive.org/web/20130123201721im_/http://www.qrcode.com/images/bigcapa.gif.

Epic Games, Inc (n.d.) *Level Editor*. [image] [Accessed on 23 November 2017] https://docs.unrealengine.com/latest/images/Engine/UI/LevelEditor/LevelEditor_Windows.jpg.

Feature Extraction. (n.d.) Uk.mathworks.com. [Online] [Accessed on 9 November 2017] https://uk.mathworks.com/discovery/feature-extraction.html.

focus group. (n.d.) [online] In: *Merriam-Webster*. [Accessed on 23 November 2017] https://www.merriam-webster.com/dictionary/focus%20group.

focus group. (n.d.) [online] In: *Oxford Dictionaries*. [Accessed on 23 November 2017] https://en.oxforddictionaries.com/definition/us/focus_group.

Focus groups. (2006) Evalued.bcu.ac.uk. [Online] [Accessed on 23 November 2017] http://www.evalued.bcu.ac.uk/tutorial/4b.htm.

Fu, K. and Rosenfeld, A. (1976) 'Pattern Recognition and Image Processing'. *IEEE Transactions on Computers*, C-25(12) pp.1336-1346.

Gao, J., Tsao, H. and Wu, Y. (2003) *Testing and quality assurance for component-based software*. Boston: Artech House, p.27.

Google VR (2017) *Afraid of the Dark*. [video] [Accessed on 29 October 2017] https://www.youtube.com/watch?v=tERxcTGOnfQ.

Höllerer, T. and Feiner, S. (2004) 'Mobile Augmented Reality'. In: H. Karimi and A. Hammad, ed., *Telegeoinformatics: Location-Based Computing and Services*. P.190.

Ikea (2013) *Place IKEA furniture in your home with augmented reality*. [video] [Accessed on 15 October 2017] https://www.youtube.com/watch?v=vDNzTasuYEw.

interview bias. (n.d.) Encyclopedia.com. [Online] [Accessed on 23 November 2017] http://www.encyclopedia.com/social-sciences/dictionaries-thesauruses-pictures-and-press-releases/interview-bias.

interviewer bias. (n.d.) BusinessDictionary.com. [Online] [Accessed on 23 November 2017] http://www.businessdictionary.com/definition/interviewer-bias.html.

iOS 11. (2017) Apple (United Kingdom). [Online] [Accessed on 29 October 2017] https://www.apple.com/uk/ios/ios-11/.

Kent, S. (2001) The ultimate history of video games. Roseville, Calif.: Prima Publ.

Knauss, D. (2016) 'Why Unity Will Power the Future of Virtual and Augmented Reality'. *Presence Press*. [Blog] [Accessed on 26 October 2017] https://presencepress.presencepg.com/why-unity-will-power-the-future-of-virtual-and-augmented-reality-6e79960fbce4.

Langshaw, M. (2011) Feature: The Rise and Fall of Arcade Gaming. Digital Spy. [Online] [Accessed on 16 April 2018] http://www.digitalspy.com/gaming/news/a328858/feature-the-rise-and-fall-of-arcade-gaming/.

Lerner, K. and Lerner, B. (2013) *Computer Sciences*. 2nd ed. Farmington Hills: Gale, Cengage Learning, pp.41-45.

Locklear, M. (2017) *Snapchat's AR Lenses are the newest tool for ads*. Engadget. [Online] [Accessed on 16 October 2017] https://www.engadget.com/2017/09/28/snapchat-ar-lenses-tool-ads/.

MacLeod, K. (2016) *Rainbows*. [Online] [Accessed on 5 April 2018] http://incompetech.com/music/royalty-free/mp3-royaltyfree/Rainbows.mp3.

McDonald, E. (2017) The Global Games Market 2017 | Per Region & Segment | Newzoo. Newzoo. [Online] [Accessed on 16 April 2018] https://newzoo.com/insights/articles/the-global-games-market-will-reach-108-9-billion-in-2017-with-mobile-taking-42/.

Melissinos, C. and O'Rourke, P. (2012) The Art of Video Games From Pac-Man to Mass Effect. New York: Welcome Enterprises, Inc.

Milgram, P., Takemura, H., Utsumi, A. and Kishino, F. (1994) *Augmented reality: A class of displays on the reality-virtuality continuum. Telemanipulator and Telepresence Technologies*. [ebook] [Accessed on 8 October 2017] http://wiki.commres.org/pds/Project_7eNrf2010/_5.pdf.

Mixed Reality. (2013) *Learning & Leading with Technology*, 40(5) p.10.

Murphy, B. (2011) *Alphabetized Cassette Tapes*. [True Type Font].

Nederhof, A. (1985) 'Methods of coping with social desirability bias: A review'. *European Journal of Social Psychology*, 15(3) pp.263-280.

Newzoo (2017) 2017 Global Games Market Q4 2017 Update. [image] [Accessed on 16 April 2018] https://newzoo.com/wp-content/uploads/2016/03/Newzoo_2017_Global_Games_Market_Per_Segment_October_2017-1024x576.png.

Nielsen, J. (2012) *Thinking Aloud: The #1 Usability Tool*. Nielsen Norman Group. [Online] [Accessed on 23 November 2017] https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/.

Oculus. (2017) Oculus.com. [Online] [Accessed on 7 October 2017] https://www.oculus.com/.

Pattern Recognition. (n.d.) Uk.mathworks.com. [Online] [Accessed on 9 November 2017] https://uk.mathworks.com/discovery/pattern-recognition.html.

PlayStation®VR. (2017) Playstation. [Online] [Accessed on 8 October 2017] https://www.playstation.com/en-gb/explore/playstation-vr/.

PTC Announces a Major New Release to Vuforia Augmented Reality Platform. (2017) Ptc.com. [Online] [Accessed on 29 October 2017] https://www.ptc.com/en/news/2017/vuforia-7.

QR Code features. (2013) Web.archive.org. [Online] [Accessed on 29 October 2017] https://web.archive.org/web/20130129064920/http://www.qrcode.com/en/qrfeature.html.

*Raycasting Beginner Physics*. (n.d.) [video] Unity Technologies.

Robustelli, D. (2017) *Whac A Mole AR/MR HoloLens prototype*. [video] [Accessed on 12 November 2017] https://www.youtube.com/watch?v=NF9ZnSCaCso.

Rouse, M. (2016) *What is augmented reality (AR)?*. WhatIs.com. [Online] [Accessed on 6 October 2017] http://whatis.techtarget.com/definition/augmented-reality-AR.

SEGA (2013) *Sonic Dash Character Select Menu*. [image] [Accessed on 5 April 2018] https://lh3.googleusercontent.com/rPLFXOYu-zt41H0TDOi3yo5uC0rXSyzQqDGc85jj5hAehxbVMw6BVNHIOVPDyGs_Xgg=w1474-h774-rw.

Sharifzadeh, S. (2017) *What is the difference between image processing and computer vision?*. Quora. [Online] [Accessed on 9 October 2017] https://www.quora.com/What-is-the-difference-between-image-processing-and-computer-vision/.

Skelly, R. (2016) *What are the main pros and cons of Unreal Engine 4 in general?*. Quora. [Online] [Accessed on 25 October 2017] https://www.quora.com/What-are-the-main-pros-and-cons-of-Unreal-Engine-4-in-general/.

Snapchat Support. (2017) Support.snapchat.com. [Online] [Accessed on 16 October 2017] https://support.snapchat.com/en-US/article/world-lenses.

Software Testing - Its Purpose and Definition. (n.d.) [Online] [Accessed on 12 April 2018] http://www.righthandtech.com/software-testing.php.

Spiegel, E., Murphy, B. and Brown, R. (2011) *Snapchat*. Standford, California: Snap Inc.

Supported Devices. (2017) Vuforia.com. [Online] [Accessed on 29 October 2017] https://vuforia.com/devices.html.

Sweeney, T. (2017) *AR And VR For The Rest of Us*. Unrealengine.com. [Online] [Accessed on 18 October 2017] https://www.unrealengine.com/en-US/blog/epic-unreal-engine-wwdc-2017.

Systrom, K. and Krieger, M. (2010) *Instagram*. Facebook.

Tarantola, A. (2012) *How QR Codes Work and Why They Suck So Hard*. Gizmodo.com. [Online] [Accessed on 8 November 2017] https://gizmodo.com/5969312/how-qr-codes-work-and-why-they-suck-so-hard.

thefsoundman (2011) *Punch 02.* [Online] [Accessed on 6 April 2018] https://freesound.org/people/thefsoundman/sounds/118513/download/118513__thefsoundman__punch-02.wav.

The Guardian (2016) *The time commitment of a Pokébattle may be inconvenient mid-stroll*. [image] [Accessed on 10 November 2017] https://i.guim.co.uk/img/media/607d87da55b3804a0e54448809be7b6aaa029fad/0_232_750_449/master/750.jpg?w=1920&q=55&auto=format&usm=12&fit=max&s=eb21231be85c741f41fd509052233d34.

The MathWorks, Inc (n.d.) *Detecting moving objects by classifying image pixels in into foreground (white pixels) and background (black pixels) using Gaussian mixture models*. [image] [Accessed on 9 November 2017] https://uk.mathworks.com/content/mathworks/uk/en/discovery/pattern-recognition/jcr:content/mainParsys/image_2.adapt.full.high.jpg/1508999469273.jpg.

The MathWorks, Inc (n.d.) *Steps in the face recognition workflow*. [image] [Accessed on 9 November 2017] https://uk.mathworks.com/content/mathworks/uk/en/discovery/face-recognition/jcr:content/mainParsys/image_0.adapt.full.high.jpg/1509001333856.jpg.

Think Aloud Testing. (n.d.) Usabilitybok.org. [Online] [Accessed on 23 November 2017] http://www.usabilitybok.org/think-aloud-testing.

Unity ARKit Plugin. (2017) Unity Technologies.

Unity Technologies (n.d.) *Editor Breakdown*. [image] [Accessed on 23 November 2017] https://docs.unity3d.com/uploads/Main/Editor-Breakdown.png.

Unity Technologies (n.d.) *Vuforia Image Targets*. [image] [Accessed on 29 October 2017] https://unity3d.com/profiles/unity3d/themes/unity/images/pages/partners/vuforia/image.jpg.

Unity - Google ARCore partner page. (2017) Unity. [Online] [Accessed on 29 October 2017] https://unity3d.com/partners/google/arcore.

Unity - Unity - Multiplatform - VR-AR. (n.d.) Unity. [Online] [Accessed on 26 October 2017] https://unity3d.com/unity/features/multiplatform/vr-ar.

Unity - Manual: Prefabs. (2017) Docs.unity3d.com. [Online] [Accessed on 24 January 2018] https://docs.unity3d.com/Manual/Prefabs.html.

Unity - Scripting API: Bounds.size. (2017) Docs.unity3d.com. [Online] [Accessed on 24 January 2018] https://docs.unity3d.com/ScriptReference/Bounds-size.html.

Unity - Scripting API: Physics.Raycast. (2017) Docs.unity3d.com. [Online] [Accessed on 25 February 2018] https://docs.unity3d.com/ScriptReference/Physics.Raycast.html.

Unreal Engine (2017) *Wingnut AR Unreal Engine Demo on iOS | WWDC 2017*. [video] [Accessed on 18 October 2017] https://youtu.be/S14AVwaBF-Y.

Unreal Engine 4 For Unity Developers. (n.d.) Docs.unrealengine.com. [Online] [Accessed on 25 October 2017] https://docs.unrealengine.com/latest/INT/GettingStarted/FromUnity/index.html.

USA Today (2017) *Yes, that's a dancing hot dog on my computer*. [image] [Accessed on 16 October 2017] https://www.gannett-cdn.com/-mm-/0f8d88b186dcce3e7ea0ff39cdfb7c6fd31cc00e/c=41-0-673-1129&r=583&c=0-0-580-1031/local/-/media/2017/07/11/USATODAY/USATODAY/636353816636234643-snapchat-hotdog.jpg.

Vienna, G. and Tappeiner, E. (2017) *Vuforia4Unreal*. Codeflügel GmbH.

Virtual Reality vs. Augmented Reality. (2015) *Augment News*. [Blog] [Accessed on 7 October 2017] http://www.augment.com/blog/virtual-reality-vs-augmented-reality/.

Vuforia Developer Portal. (2017) Developer.vuforia.com. [Online] [Accessed on 29 October 2017] https://developer.vuforia.com/.

Wareable (2015) *Soldiers using virtual reality for training*. [image] [Accessed on 8 October 2017] https://static2.wareable.com/media/imager/7023-b877618af035d2eb14fd42df58db5198.jpg.

whack-a-mole Gayla's highscore = 140. (2005) [image] [Accessed on 11 November 2017] https://www.flickr.com/photos/79511840@N00/18984918.

whack-a-mole. (n.d.) Oxford Dictionaries | English. [Online] [Accessed on 11 November 2017] https://en.oxforddictionaries.com/definition/us/whack-a-mole.

Whiting, N. (2017) 'Get Started With Google ARCore Using Unreal Engine Today'. *Unreal Engine*. [Blog] [Accessed on 29 October 2017] https://www.unrealengine.com/en-US/blog/get-started-with-google-arcore-using-unreal-engine-today.

Whiting, N. (2017) *Getting Started with UE4 and ARKit*. Unrealengine.com. [Online] [Accessed on 22 October 2017] https://www.unrealengine.com/en-US/blog/getting-started-with-ue4-and-arkit.

Wilson, J. (2017) 'Unreal Engine 4.18 Released!'. *Unreal Engine*. [Blog] [Accessed on 29 October 2017] https://www.unrealengine.com/en-US/blog/unreal-engine-4-18-released.

Wired (2012) *Nokia City Lens' augmented reality mode overlays businesses on a live camera view*. [image] [Accessed on 8 October 2017] https://media.wired.com/photos/5932f940d80dd005b42b0b3f/master/w_799,c_limit/120509-NOKIA-CITY-LENS-004edit.jpg.

Wong, M. (2017) 'Programmer Perspective: Switching from Unity to Unreal Engine 4'. *Dreamsail Games*. [Blog] [Accessed on 25 October 2017] https://dreamsailgames.com/blog/2017/5/4/unreal-engine-versus-unity.

# 8 APPENDICES

## 8.1 TERMS OF REFERENCE

### 8.1.1 Learning Outcomes

- Apply design principles, use design methodologies, and produce documentation for game specifications using industry standard approaches
- Understand the key technical problems and issues in three-dimensional modelling, rendering, and animation, and be able to monitor new developments in these areas
- Implement systems and frameworks that demonstrate that they are a competent user of the mathematical tools and techniques used in three-dimensional computer games and in game physics
- Demonstrate effective communication, decision making, and creative problem solving skills, and identify appropriate practices within a professional, legal, and ethical framework

### 8.1.2 Project Background

With the increase in popularity of both virtual and augmented reality in video games, especially for mobile devices, as well as the rising popularity of 'on-the-go' games, the Augmented Reality 'Whack-A-Mole' project looks to bring the two concepts together to create a new video game for mobile devices. The game would target those interested in augmented reality for games, as well as 'casual' gamers.

### 8.1.3 Aim

The project aims to design an augmented reality video game similar to the popular arcade game 'Whack-A-Mole' which can be implemented primarily on mobile devices, that involves the development and implementation of an Augmented Reality (AR) framework that can be used to display creatures from the game.

### 8.1.4 Objectives

- Research
    - Existing augmented reality software
    - Differences in Unity and Unreal Engine for use in AR game development
    - Augmented reality and its use in relation to video games
    - Computer Vision
- Testing and Feedback
    - Mechanics with focus group made up of target audience
    - Feedback on usability of the application
    - AR framework tested on mobile device
- Development
    - Game mechanics
    - Augmented reality framework – must work with at least one 3D model

- Animations for use with augmented reality
- User interface

### 8.1.5 Problems

| Potential Problem | Solution |
|---|---|
| Focus group may not be available to provide feedback and testing of mechanics | Provide advance notice and extend invitation to more people |
| Work may be lost due to power cuts, malfunctions etc | Maintain backups as well as previous version copies |

*Table 8.1 - Potential problems and solutions*

### 8.1.6 Timetable and Deliverables

| Task | Planned Submission Date | Task Description |
|---|---|---|
| **Literature Review** | 05/11/2017 | Analyse and evaluate existing concepts, products, and methods related to the project:<br>• Existing AR apps<br>• Computer vision<br>• Unity vs UE4<br>• AR in video games |
| **Initial Product Design** | 15/12/2017 | Basic design work that may be revised several times during the prototype development stage |
| **Prototype Development** | 07/01/2018 | • Analyse the chosen game engine and AR API functionality for use in the project<br>• Design user interface and pseudocode for mole behaviours<br>• Build mole prototype |
| **Initial Testing** | 04/02/2018 | Test prototype to ensure basic functionality works as expected, including the AR framework, before expanding onto graphical assets |
| **Evaluation of Design** | 16/02/2018 | Analyse best method for evaluating the project, including developing a basic outline of the evaluation process |
| **Report Outline** | 23/02/2018 | Outline the chapter headings and subheadings to be used in the report |
| **Final Development** | 04/03/2018 | Complete the project in the chosen game engine, including the addition of graphical assets for the user interface where appropriate |
| **Draft Presentation Slides** | 09/03/2018 | Create slides for a draft presentation that may be later revised for the final presentation |

| | | |
|---|---|---|
| **Complete Evaluation** | 20/04/2018 | • Finalise evaluation process, including any form of black/white box testing, user surveys, focus groups, etc<br>• Evaluate product with the finalised evaluation methods<br>• Review and conclude results |
| **Final Presentation** | 27/04/2018 | Compile final slides for presentation |
| **Report Submission** | 27/04/2018 | Compile final report for submission |

*Table 8.2 - Timetable of tasks and deliverables*

### 8.1.7 Required Resources

- Game engine
- AR-compatible mobile device
- Third-party APIs for AR development

## 8.2 ETHICS AND RISK ASSESSMENT

**ETHICS CHECKLIST**

This checklist must be completed **before** commencement of **any** research project. This includes projects undertaken by **staff and by students as part of a UG, PGT or PGR programme**. Please attach a Risk Assessment.

Manchester Metropolitan University

Please also refer to the University's Academic Ethics Procedures; Standard Operating Procedures and the University's Guidelines on Good Research Practice

| | |
|---|---|
| **Full name and title of applicant:** | Jasmine Dodson |
| **University Telephone Number:** | |
| **University Email address:** | jasmine.a.dodson@stu.mmu.ac.uk |
| **Status:**<br><br>All staff and students involved in research are strongly encouraged to complete the Research Integrity Training which is available via the Staff and Research Student Moodle areas | Undergraduate Student ☑<br>Postgraduate Student: Taught ☐<br>Postgraduate Student: Research ☐<br>Staff ☐ |
| **Department/School/Other Unit:** | Computing, Maths & Digital Technology |
| **Programme of study (if applicable):** | Computer Games Technology |
| **Name of DoS/Supervisor/Line manager:** | |
| **Project Title:** | Augmented Reality Card Game |
| **Start & End date (cannot be retrospective):** | 01/10/17 - 31/04/18 |
| **Number of participants (if applicable):** | |
| **Funding Source:** | |

**Brief description of research project activities (300 words max):**

The project aims to design a collectable card game/trading card game (CCG/TCG) which can be implemented primarily on mobile devices, that involves the development and implementation of an Augmented Reality (AR) framework that can be used to display creatures from the game. The game should have potential for expansions which can be purchased within the game using "micro-transactions".

| | YES | NO |
|---|---|---|
| **Does the project involve NHS patients or resources?**<br>If 'yes' please note that your project may need NHS National Research Ethics Service (NRES) approval. Be aware that research carried out in a NHS trust also requires governance approval.<br><br>Click here to find out if your research requires NRES approval<br><br>Click here to visit the National Research Ethics Service website<br><br>To find out more about Governance Approval in the NHS click here | ☐ | ☑ |
| **Does the project require NRES approval?** | ☐ | ☑ |
| If yes, has approval been granted by NRES?<br>Attach copy of letter of approval. Approval cannot be granted without a copy of the letter. | ☐ | ☐ |

| NB Question 2 should only be answered if you have answered YES to Question 1. All other questions are mandatory. | YES | NO |
|---|---|---|
| 1. Are you are gathering data from people? | ☑ | ☐ |
| For information on why you need informed consent from your participants please click here | | |
| 2. If you are gathering data from people, have you: | ☐ | ☐ |
| a. attached a participant information sheet explaining your approach to their involvement in your research and maintaining confidentiality of their data? | ☐ | ☑ |
| b. attached a consent form? (not required for questionnaires) | ☐ | ☑ |
| Click here to see an example of a participant information sheet and consent form | | |
| 3. Are you gathering data from secondary sources such as websites, archive material, and research datasets? | ☑ | ☐ |
| Click here to find out what ethical issues may exist with secondary data | | |
| 4. Have you read the guidance on data protection issues? | ☑ | ☐ |
| a. Have you considered and addressed data protection issues – relating to storing and disposing of data? | ☑ | ☐ |
| b. Is this in an auditable form? (can you trace use of the data from collection to disposal) | ☐ | ☑ |
| 5. Have you read the guidance on appropriate research and consent procedures for participants who may be perceived to be vulnerable? | ☑ | ☐ |
| a. Does your study involve participants who are particularly vulnerable or unable to give informed consent (e.g. children, people with learning disabilities, your own students)? | ☐ | ☑ |
| 6. Will the study require the co-operation of a gatekeeper for initial access to the groups or individuals to be recruited (e.g. students at school, members of self-help group, nursing home residents)? | ☐ | ☑ |
| Click for an example of a PIS and information about gatekeepers | | |
| 7. Will the study involve the use of participants' images or sensitive data (e.g. participants personal details stored electronically, image capture techniques)? | ☐ | ☑ |
| Click here for guidance on images and sensitive data | | |
| 8. Will the study involve discussion of sensitive topics (e.g. sexual activity, drug use)? | ☐ | ☑ |
| Click here for an advisory distress protocol | | |
| 9. Could the study induce psychological stress or anxiety in participants or those associated with the research, however unlikely you think that risk is? | ☐ | ☑ |
| Click here to read about how to deal with stress and anxiety caused by research procedures | | |
| 10. Will blood or tissue samples be obtained from participants? | ☐ | ☑ |
| Click here to read how the Human Tissue Act might affect your work | | |
| 11. Is your research governed by the Ionising Radiation (Medical Exposure) Regulations (IRMER) 2000? | ☐ | ☑ |
| Click here to learn more about IRMER | | |
| 12. Are drugs, placebos or other substances (e.g. food substances, vitamins) to be administered to the study participants or will the study involve invasive, intrusive or potentially harmful procedures of any kind? | ☐ | ☑ |
| Click here to read about how participants need to be warned of potential risks in this kind of research | | |
| 13. Is pain or more than mild discomfort likely to result from the study? Please attach the pain assessment tool you will be using. | ☐ | ☑ |

| | | |
|---|---|---|
| Click here to read how participants need to be warned of pain or mild discomfort resulting from the study and what do about it. | | |
| 14. Will the study involve prolonged or repetitive testing or does it include a physical intervention? | ☐ | ☑ |
| Click here to discover what constitutes a physical intervention and here to read how any prolonged or repetitive testing needs to managed for participant wellbeing and safety | | |
| 15. Will participants to take part in the study without their knowledge and informed consent? If yes, please include a justification. | ☐ | ☑ |
| Click here to read about situations where research may be carried out without informed consent | | |
| 16. Will financial inducements (other than reasonable expenses and compensation for time) be offered to participants? | ☐ | ☑ |
| Click here to read guidance on payment for participants | | |
| 17. Is there an existing relationship between the researcher(s) and the participant(s) that needs to be considered? For instance, a lecturer researching his/her students, or a manager interviewing her/his staff? | ☐ | ☑ |
| Click here to read guidance on how existing power relationships need to be dealt with in research procedures | | |
| 18. Have you undertaken Risk Assessments for each of the procedures that you are undertaking? | ☐ | ☑ |
| 19. Is any of the research activity taking place outside of the UK? | ☐ | ☑ |
| 20. Does your research fit into any of the following security sensitive categories:<br>• commissioned by the military<br>• commissioned under an EU security call<br>• involve the acquisition of security clearances<br>• concerns terrorist or extreme groups<br>If Yes, please complete a Security Sensitive Information Form | ☐ | ☑ |

I understand that if granted, this approval will apply to the current project protocol and timeframe stated. If there are any changes I will be required to review the ethical consideration(s) and this will include completion of a 'Request for Amendment' form.

☐ have attached a Risk Assessment
☐ have attached an Insurance Checklist

If the applicant has answered **YES** to **ANY** of the questions **5a – 17** then they must complete the MMU Application for Ethical Approval.

Signature of Applicant: Digitally signed by Jasmine Dodson Date: 2017.06.23 12:45:16 +01'00' Date: 21/06/17 (DD/MM/YY)

***Independent Approval* for the above project is (please check the appropriate box):**
***Granted***
☑ I confirm that there are no ethical issues requiring further consideration and the project can commence.

***Not Granted***
☐ I confirm that there are ethical issues requiring further consideration and will refer the project protocol to the Faculty Research Group Officer.

Signature: Kevin Tan Digitally signed by Kevin Tan Date: 2017.06.23 17:28:42 +01'00' Date: 23/06/17 (DD/MM/YY)

Print Name: KEVIN TAN Position: Senior Lecturer

**Approver: Independent Scrutiniser for UG and PG Taught/ PGRs RD1 Scrutiniser/ Faculty Head of Ethics for staff.**

**The MANCHESTER METROPOLITAN UNIVERSITY**
**Faculty of Science and Engineering**

# RISK ASSESSMENT COVER SHEET

**REFERENCE NUMBER**: NPC / 090517 / JDE1.49

**SCHOOL: Computing, Mathematics & Digital Technology**

**TITLE OF WORK:** CMT Projects involving software development and acceptability testing

**LOCATION OF WORK:** John Dalton Building computing facilities, computers at student's own home etc.

**INTENDED ACTIVITIES** (attach methods sheets (e.g. standard operating practices) and work schedules to this form):

General use of computers to develop and test software, using other participants and feedback sheets / programs. Method sheets and work schedules not applicable.

**PERSONS AT RISK** (list names of all individuals (including status e.g. staff/student), and/or unit(s) / course(s) undertaking the activity. For students please indicate course and level, for staff give contact email / phone number):

Undergraduate students and the participants they recruit.

**HAZARDS** (provide a summary of the hazards anticipated and attach detailed assessments with appropriate risk control methods to this form):

Repetitive Strain Injury – work related upper limb disorder

Back injury resulting from improper posture

Eye strain

Fatigue Stress

Possible risk from 240v electrical mains supply

Normal dangers of moving about the University to attend testing sessions.

*Are these hazards necessary in order to achieve the objectives of the activity?*

**Yes**

**Hazard Rating** (delete as appropriate): **Low**

**HAZARDOUS SUBSTANCES/MATERIALS USED AND HAZARD CLASSIFICATION**
**(**appropriate COSHH data sheets / risk assessments must be attached to this form**):**
**ALL CONTAINERS OF HAZARDOUS SUBSTANCES SHOULD BEAR CORRECT HAZARD WARNING LABELS**.

| **NAME OF MATERIAL** *Please provide also approximate quantity and concentration if applicable.* | **HAZARD CLASS** | **HAZARD LABEL** | **DISPOSAL** *Hazardous materials must not be removed from laboratories. List disposal arrangements for* all materials listed below *in the location where* the work will be |
|---|---|---|---|

| | | | |
|---|---|---|---|
| | | | carried out: |
| | | | |

**RISK CONTROL METHODS** (provide a summary of the hazards anticipated and attach detailed assessments with appropriate risk control methods to this form):

The hazards identified above are controlled by:

Facilities review when laboratories are commissioned
Induction session on H&S given to students by Technical Services Manager
School H&S information given in Student handbook
Posters in laboratories
PAT testing of equipment after three years
Annual H&S inspections

*The laboratory workstations, whilst not legally required to be DSE compliant, (the continuous usage is too low to present risk) are fully compliant with current legislation. Monitors and keyboards are adjustable, chairs are adjustable and the lighting designed for both computer usage and associated reading activity. In each laboratory, there is an adjustable desk, suitable for wheelchair users, usually located in the next to the door.*

**Hazard Rating with Control Methods** (delete as appropriate): **Low**

*Will any specific* **training** *be required (if* YES *give details)?* N/A

*Are there any specific* **first aid** *issues (if* YES *give details)?* N/A

**PROCEDURE FOR EMERGENCY SHUT-DOWN** (if applicable):

In the event of fire, flood or other emergency, evacuation of the laboratory would take place and the technical staff would subsequently make an assessment of the necessity of switch-off. As overall system control is vested in a separate server room, there would be little physical harm to any device in directly cutting the power to the mains for each individual lab.

Re-start of the lab may present problems of a technical nature but would not affect the personal safety or health of any individual.

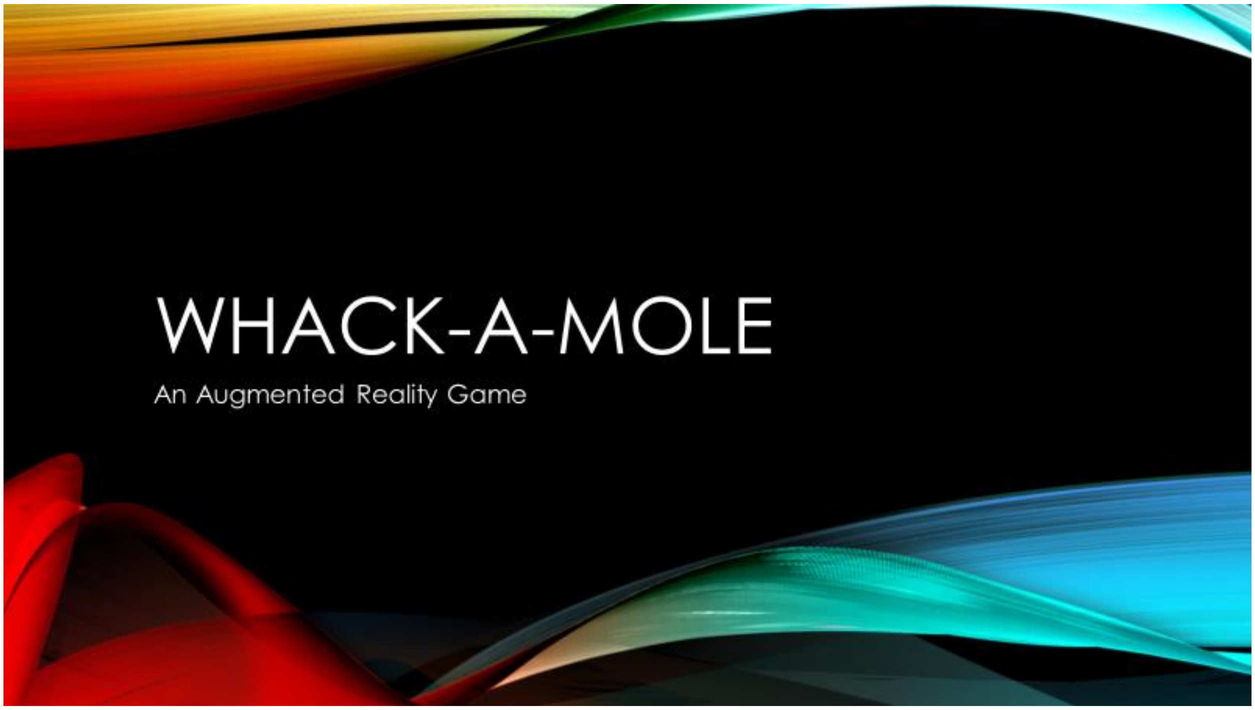
**IF OFF-SITE INDICATE ANY OTHER ISSUES** (e.g. associated with: individual's health and dietary requirements (obtain off-site health forms for all participating individuals and indicate where this information will be located); social activities, transportation, ID requirements; permissions for access and sampling).

**Not applicable – this form applies only to the laboratories listed**

| | **NAME** | **STAFF/STUDENT No.** | **DATE** |
|---|---|---|---|
| **Originator** | Nicholas Costen | 01900261 | 171016 |
| **Supervisor** | N/A | | |
| **Technical Manager** | | | |
| **Divisional / School Health and Safety Coordinator** (p.p. HoS) | Digitally signed by Nicholas Costen<br>DN: cn=Nicholas Costen, o=MMU, ou=SCMDT, email=n.costen@mmu.ac.uk, c=GB<br>Date: 2017.05.24 16:34:34 +01'00' | | |

**DATE TO BE REVIEWED BY: September 2018**

## 8.3 PRESENTATION SLIDES

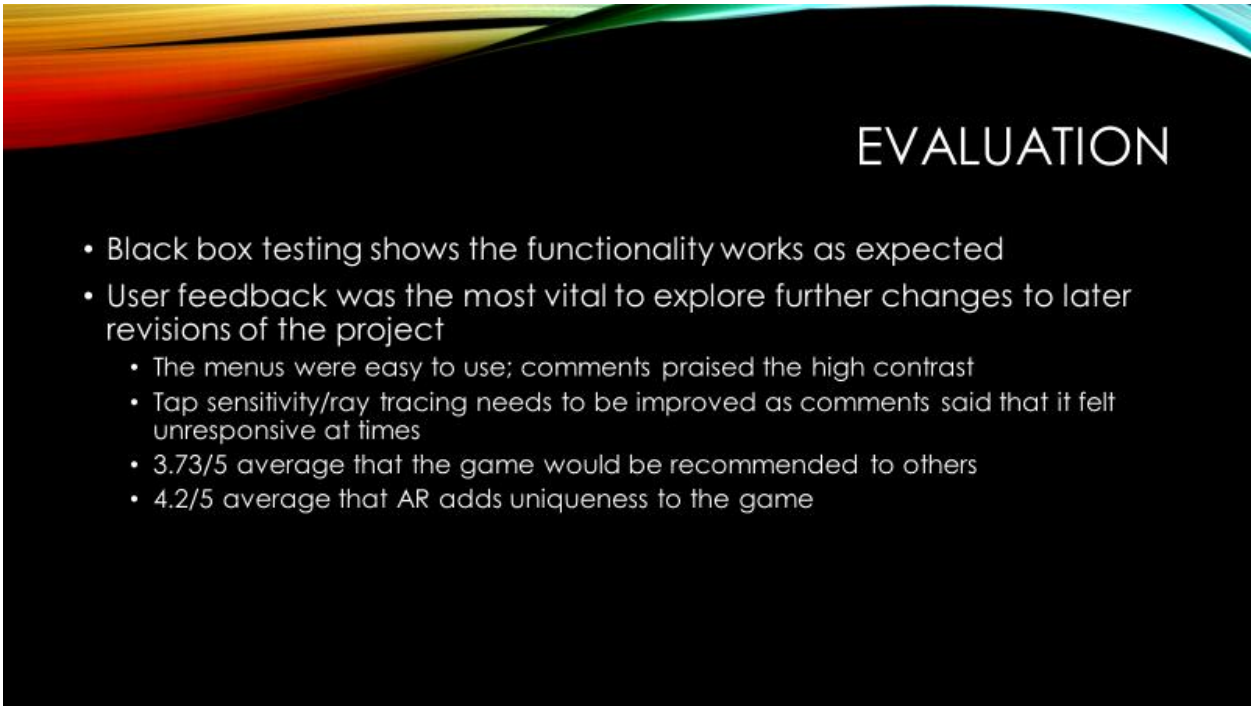



## EXISTING SYSTEMS

- Existing mobile apps
  - Snapchat
  - Pokémon Go
- Game engines with in-built augmented reality development support
  - Unreal Engine 4
  - Unity 5
  - ARKit
  - ARCore
  - Vuforia

## SOLUTION

- Unity 5 to build and script the game
- Vuforia for augmented reality
- Developed and tested on PC
  - Also tested on a mobile device


EVALUATION
• Black box testing shows the functionality works as expected
• User feedback was the most vital to explore further changes to later revisions of the project
• The menus were easy to use; comments praised the high contrast
• Tap sensitivity/ray tracing needs to be improved as comments said that it felt unresponsive at times
• 3.73/5 average that the game would be recommended to others
• 4.2/5 average that AR adds uniqueness to the game


CONCLUSION
• Use of augmented reality in games is on the rise
• Game engines have in-built support for development
• Further improvements include removing the need for an image target
• With a dedicated team the project could become more polished
• Specific custom art assets
• Online functionality
• e.g. online scoreboard
• Play with friends


## 8.4 OneDrive Directory

The entire Unity project folder, including the source code, scene files, apk file, and additional assets can be found in a zip file at the below OneDrive link.

https://stummuac-my.sharepoint.com/:u:/g/personal/15081839_stu_mmu_ac_uk/EYSB0hmmpClHoAnwQZnd-7IBaM7QrBZylQzQZE2zPOeiLQ

## 8.5 Miscellaneous

### 8.5.1 Questionnaire Response Page

https://forms.office.com/Pages/ResponsePage.aspx?id=UPs_KAujjEiQ9M2uT3rm0R2MzljpaMlJnUOsEln48jVUOFpONjcyMFk4NjRNSE1VS0U4MkdaMTJJTC4u

### 8.5.2 Questionnaire Summary Page

https://forms.office.com/Pages/AnalysisPage.aspx?id=UPs_KAujjEiQ9M2uT3rm0R2MzljpaMlJnUOsEln48jVUOFpONjcyMFk4NjRNSE1VS0U4MkdaMTJJTC4u&AnalyzerToken=3UZlykxL74di1S9XFxRsYmhiqRe066u1