

A Low Cost Virtual Reality Interface for Educational Games



Tashiv Sewpersad

Department of Computer Science

University of Cape Town

Supervisor

Professor James Gain

In fulfilment of the requirements for the degree of *Masters of Science in Computer Science*

March 12, 2021

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

Table of Contents

Chapter 1: Introduction	5
1.1. Research Question	6
1.2. Objectives and Contributions	7
1.3. Overview of Thesis.....	7
Chapter 2: Background.....	8
2.1. Virtual Reality	8
2.2. Virtual Reality Interfaces	9
2.2.1. Gaze	9
2.2.2. Tilt	9
2.2.3. Computer Vision	10
2.2.4. Magnetic Switch.....	10
2.2.5. External Electronics.....	10
Chapter 3: System Overview	12
3.1. Introduction	13
3.2. System Design	13
3.2.1 Frontend Application System	13
3.2.2. Backend Computer Vision System.....	14
Chapter 4: Backend Computer Vision System.....	16
4.1. Introduction	16
4.2. Related Work.....	17
4.2.1. Key Point Detection Algorithms	17
4.2.2. Key Point Description Algorithms	18
4.2.3. Tracking Algorithms	19
4.3. Implementation Details	19
4.3.1. Marker Generator	20
4.3.2. Image Detection Process	21
4.3.3. Image Tracking Extension.....	22
4.4. Evaluating the Pipeline.....	22
4.4.1. Performance of the Image Detection Pipeline	22
4.4.2. Performance of the Extended System.....	25
Chapter 5: Front-end Virtual Reality Interface.....	27
5.1. Immersive Serious Games	27
5.2. Iterative Design Methodology	27

5.3. Low-Fidelity Paper Prototype	29
5.4. High-Fidelity 3D-Printed Prototype	31
5.4.1. Physical Interface Development	31
5.5. Fauna and Flora Identification Game	37
5.5.1. Game Design	37
5.5.2. Controller Implementation in Software	39
Chapter 6: Experimental Design	41
6.1. Introduction	41
6.2. Experimental Design Overview	42
6.3. Task Design	44
6.4. Measures	45
6.5. Experimental Procedure	46
6.6. Extension to Enable Testing under Covid-19 Pandemic	46
Chapter 7: Results & Discussion	49
7.1. Demographics	49
7.2. Game Experience Questionnaire Results	51
7.3. Performance Metrics	54
7.3.1. Battery Consumption	54
7.3.2. Time Performance	55
7.3.3. Score Performance	58
7.3.4. Controller Usage	60
7.4. Qualitative Feedback	61
7.4.1. Electronic Controller Feedback	62
7.4.2. Vision Controller Feedback	63
7.4.3. Game Experience Feedback	65
7.5. Conclusions	66
Chapter 8: Conclusions & Future Work	67
8.1. Conclusions	67
8.2. Future Work	68
References	69
Appendix A: Educational Game Assets	73
Appendix B: Evaluation Documentation	74
Appendix C: User Feedback	82

Table of Figures

Figure 1.a	VR Headset Examples.....	5
Figure 1.b	User Centred Design Flow.....	7
Table 2.a	Mobile VR Interface Types.....	11
Figure 3.a	Vision Controller System Diagram.....	12
Figure 4.a	Tracking Surface Examples.....	16
Table 4.b	Marker Generator Placement Algorithms.....	20
Figure 4.c	Marker Generator Samples.....	21
Figure 4.d	Example Scene Frames for ORB.....	23
Table 4.e	Image Matching Test Ranges.....	23
Table 4.f	Predication Accuracy for SIFT, SURF, ORB.....	23
Figure 4.g	Predication Performance for SIFT, SURF, ORB.....	24
Table 4.h	Algorithm Timings for SIFT, SURF, ORB.....	25
Table 4.i	Predication Accuracy for ORB, ORB + Tracking.....	25
Figure 4.j	Predication Performance for ORB, ORB + Tracking.....	26
Figure 4.k	Prediction Error Example.....	26
Figure 5.a	Paper Prototype.....	29
Table 5.b	Interface Components of Paper Prototype.....	29
Table 5.c	Feedback for Interface Components of Paper Prototype.....	30
Figure 5.d	Iteration 1 of High Fidelity Prototype.....	32
Figure 5.e	Iteration 2 of High Fidelity Prototype.....	33
Table 5.f	Selection Wheels for Iteration 2.....	34
Figure 5.g	Iteration 3 of High Fidelity Prototype.....	35
Figure 5.h	Haptic Feedback extension of Iteration 3 of High Fidelity Prototype.....	36
Figure 5.i	Flipped controller extension of Iteration 3 of High Fidelity Prototype.....	36
Figure 5.j	VR Game Terrains.....	37
Figure 5.k	VR Game Views.....	38
Figure 5.l	Vision Controller Interface of VR Game.....	39
Figure 5.m	Electronic Controller Interface of VR Game.....	40
Figure 6.a	Evaluated Interfaces.....	41
Table 6.b	Pre-experiment Questionnaire Fields.....	42
Figure 6.c	VR Game Screenshot.....	43
Figure 6.d	VR Game Terrains by Set.....	43
Table 6.e	Mitigation Strategies for Erroneous Data.....	44
Figure 6.f	Packaging for Remote Evaluation.....	47
Figure 6.g	Software Enhancements for Remote Evaluation.....	47
Graph 7.a	Fields of Study of Participants.....	49
Graph 7.b	Participant Experience Levels.....	50
Table 7.c	Normality of GEQ Metrics.....	51
Table 7.d	Statistical Difference of GEQ Metrics.....	52
Figure 7.e	Box & Whisker Plots of GEQ Metrics.....	53
Table 7.f	GEQ Metrics by Class.....	53
Graph 7.g	Batter Consumption across Controller Types.....	54
Graph 7.h	Evaluation Times across Controllers.....	55
Figure 7.i	Average Time Spent per Question across Controller Orders.....	56
Figure 7.j	Average Time Spent per Question across Controllers.....	57
Figure 7.k	Average Score per Question across Controller Orders.....	58
Figure 7.l	Average Score per Question across Controllers.....	59
Figure 7.m	Average Score per Question across Controllers, Segmented.....	59
Table 7.n	Statistical Differences across Controller Questions.....	60
Figure 7.o	Comparison of Magnification Usage Across Controllers.....	61
Figure 7.p	Comment Count per Heuristic for Electronic Controller.....	62
Figure 7.q	Comment Count per Heuristic for Vision Controller.....	63
Figure 7.r	Comment Count per Heuristic for Game.....	65

Chapter 1:

Introduction

Student motivation and engagement are important factors in learning environments as they influence knowledge retention and perseverance in learning. However, Dicheva et al. [12], suggest that schools are facing challenges in motivating learners and keeping them engaged. They go on to discuss how serious games, digital video games that aim to both educate and entertain [3], may be a solution to this issue. Serious games aim to be immersive, while also providing mechanisms for teaching and reinforcing knowledge [12]. Given that strong engagement has been shown to correlate with better academic performance [42], serious games should be immersive and engaging in order to be effective at teaching.

Virtual Reality (VR) is a field within Computer Graphics that has recently been revitalized by the emergence of products like Google Cardboard and the Oculus Rift. One of the aims of VR is to improve user immersion in 3D simulations [52], such as serious games, using visual, auditory and haptic elements. While this can be done in a variety of ways, systems like the Samsung Gear VR and the Oculus Rift make use of head-mounted displays (HMDs) and sensor arrays (see Figure 1.a). These headsets use data collected from their sensor arrays to track a user's head movement and update the displayed view accordingly. This gives a user the sensation of being in the 3D environment being simulated.

An example of how VR can improve immersion can be seen in the game VRun by Yoo and Kay [52]. This study shows that players feel more immersed in the VRun game when using a VR headset over a conventional computer screen. Commercial serious VR games also exist and an example can be seen in the *Expeditions*¹ mobile application by Google. This application allows users to learn about geographical landmarks in a more immersive way by enabling users to explore these landmarks as if they were there. Given VR's potential for improving user immersion, it shows promise for enhancing student engagement in the context of serious games.



Figure 1.a. These are examples of Virtual Reality Headsets and accessories. On the left is the smartphone-driven Samsung Gear VR with an electronic controller. On the right is the desktop-driven Oculus with two controllers and two base stations for tracking.

¹ Available at <https://play.google.com/store/apps/details?id=com.google.vr.expeditions>

While VR has great potential for improving student engagement, current desktop-driven VR headsets are costly to use as each user requires a high-end VR headset and a gaming-grade computer to run it. This type of system typically costs around \$2400 when using the Oculus Rift (\$1000) and a VR-ready Nvidia GTX1060 based desktop (\$1400). In contrast, smartphone-driven VR solutions like the Samsung Gear VR have a lower relative cost due to their use of smartphones as drivers. These types of systems usually cost around \$500 when using the Samsung Galaxy S7 smartphone (\$400) and the 2016 Samsung Gear VR headset (\$100). This lower cost means that for the same price, more mobile driven headsets can be deployed to a given multiuser setting (e.g., a classroom). Additionally, Amin et al. [1] show that lower fidelity smartphone-driven headsets can provide comparable immersion results to higher fidelity desktop-driven VR headsets, for games that run well on both devices.

While the application of smartphone-driven VR headsets has potential benefits for education, some research challenges remain. An issue with mobile driven VR headsets is interface choice, as the VR headset obstructs the embedded smartphone's primary user interface (i.e., the touchscreen) [44]. This choice is important because the level of immersion of a simulation is influenced by the type of interface used [20]. Currently, many interface options exist for mobile VR ranging from internal smartphone sensor usage [49] to separate external controllers [41].

After conducting a literature review (see section 2.2), it appears that computer vision (CV) based interfaces have not been widely explored in the context of mobile VR, even though they show promise as a relatively low cost and immersive interface approach. CV based interfaces may also be appropriate for multiuser simulations as they allow for shared interfaces between multiple users. This is possible because different users can track the same image targets simultaneously from different perspectives. For example, consider a VR version of chess where the chessboard is a large trackable surface. It is likely that CV based interfaces have not been widely explored due to the difficulty of balancing the computational cost of computer vision algorithms and VR rendering on current smartphone hardware. Even though this interface approach shows promise, further research is required in order to more fully understand the benefits and limitations in a mobile VR context.

1.1. Research Question

This research project aims to investigate a computer-vision based user interface for mobile VR that may be applied in multi-user educational contexts. This interface approach will make use of image recognition, as these types of interfaces have not been widely explored in the literature, even though they show great potential as an immersive interface that is both highly expressive and cost effective (See section 2.2, table 2.a). It is speculated that the processing cost of image recognition algorithms has discouraged research into these types of interfaces in the mobile VR context. However, it is believed that modern smartphones now have the processing power to support these types of interfaces as evidenced by the emergence of augmented reality (AR) technologies like ARCore² by Google and ARKit³ by Apple, which showcase the image recognition capabilities of modern smartphones in a mobile AR setting. VR differs from AR in that it requires stereoscopic rendering to enable the use of head-mounted displays (HMDs), while AR smartphone applications do not, as they do not use HMDs. This makes VR more computationally expensive to use in this case.

² See <https://developers.google.com/ar> for more information.

³ See <https://developer.apple.com/augmented-reality/> for more information.

For this project, a serious game will be implemented as doing so will allow an answer to the primary research question, “*can a computer-vision based virtual reality controller provide comparable immersion to a conventional electronic controller?*” In answering this research question, this project will also evaluate the strengths and weaknesses of this approach when compared to a conventional electronic controller. Given that the literature reviewed suggests that mobile VR has a good balance between quality and cost, a smartphone-driven Samsung Gear VR headset (left image of Figure 1.a.) will be used in this project. It is important to note that this project aims to investigate an interface approach in the context of a learning environment. Therefore, the learning application itself is not the main focus of this project, and thus the learning impact of this application will not be evaluated.

1.2. Objectives and Contributions

The overall goal of this project is to explore an image recognition based user interface for use in mobile VR. This goal can be broken down into the following sub-goals:

- To develop a computer vision pipeline that enables an Android device to track image markers using its camera.
- To iteratively (see figure 1.b) develop a computer-vision based VR interface that is as immersive as a conventional electronic controller. This entails developing both a physical controller and its software implementation.
- To develop a testing environment for the interface being explored. This environment will take the form of a serious mobile VR game that allows users to experience both interfaces being investigated.

There are also two direct software artefacts for this project:

1. A library that encapsulates the computer vision pipeline needed to realize the interface.
2. A serious VR game to be used in the evaluation of the interface developed.

1.3. Overview of Thesis

This thesis is organised into a number of chapters and the order of these chapters mirror the project’s development. Chapter 2 assesses literature related to smartphone-driven VR interfaces in order to determine where focus should be placed. Chapter 3 provides an overview for the solution being developed and chapter 4 investigates appropriate feature matching algorithms for a computer vision system. Chapter 5 details the development of the controller along with the game employed for evaluation. Chapter 6 goes on to discuss the evaluation’s design and chapter 7 analyses and discusses the results of the evaluation. Chapter 8 concludes the thesis by presenting conclusions and future work.

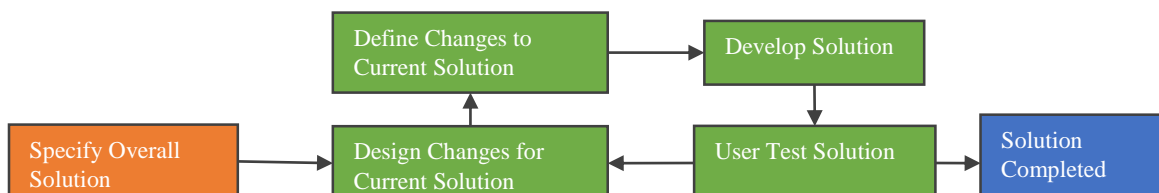


Figure 1.b. The User Centred Design Process used by this project during the development of the computer-vision based controller.

Chapter 2:

Background

This chapter contains the background information required to understand the overall context of the project. Specific background information related to the implementation of this project has been included in the chapters corresponding to those implementations.

2.1. Virtual Reality

In recent years, VR has found renewed popularity due to hardware becoming more affordable (e.g., Google Cardboard) [1, 36, 52], developer workflows becoming simpler to use (e.g., Unity 3D Game Engine) [36, 52] and improvements in graphics hardware capabilities [41]. This growth has resulted in the emergence of a spectrum of VR headsets, ranging from the low fidelity smartphone-driven Google Cardboard to the high fidelity desktop-driven Oculus Rift. While desktop-driven VR offers more capabilities, such as being able to track a user's translational movements in Euclidean space (positional tracking), smartphone-driven VR has the benefit of being more portable since tethering to an external computer is not required. This makes mobile VR more accessible [41], at the cost of being constrained by the smartphone's limited power budget, limited processing capacity and limited sensor accuracy.

This suggests that the higher-fidelity smartphone-driven Samsung Gear VR represents a good balance of cost and portability when compared to the desktop-driven Oculus Rift. The Gear VR also represents a good balance of quality and cost when compared to the lower fidelity smartphone-driven Google Cardboard [1, 41]. One potential competitor to a Gear VR based solution is the Oculus Quest 2, which can be purchased for around \$400 and is a standalone mobile VR solution. While a Gear VR solution may cost \$100 more, it can make use of smartphones already owned by students. This would provide a valuable cost reduction in classroom settings where many students already own powerful smartphones.

One challenge in VR is performance. Specifically, VR rendering requires lens and chromatic aberration corrections as well as stereoscopic rendering during the rendering phase of the application [41]. This is especially problematic for mobile devices which have a limited processing budget, when compared to their more powerful desktop counterparts. Performance issues may result in rendering delays or lag, which may cause a phenomenon known as simulator sickness. Users experiencing simulator sickness typically feel disorientated and, in extreme cases, severely nauseated [37]. Given that simulator sickness has the potential to negatively impact a user, it is important to understand its potential causes so that they can be mitigated. It is believed that the cause of simulator sickness in VR is linked to conflicts in visual (e.g., what is seen on the VR headset) and vestibular information (e.g., actual orientation and motion of a user) [30]. One of the more understood causes of simulator sickness is display delay [28], which is the latency between head movements and the rendering of scene updates. This means that VR applications should be optimized as far as possible to ensure that there is no lag in rendering. Companies like Oculus and Google both recommend 30-60 frames per second (FPS) for mobile VR applications in order to keep application responsiveness high and user visual discomfort low.

2.2. Virtual Reality Interfaces

A game's immersion is influenced by the type of interface used and generally the more suitable the interface, the more immersive the game [36]. Smartphone-driven VR presents a problem in that the headsets prevent users from accessing the embedded smartphone's touchscreen, which is the primary means of controlling the device [44]. A variety of alternate interfaces have been proposed and these are assessed in this subsection in terms of monetary cost and level of expressivity. Cost is a relevant factor in mobile VR as lower fidelity VR aims to be more accessible, especially in terms of the monetary costs [53]. The level of expressivity is the number of actions that can be undertaken with a given interface and this is also a relevant factor as certain learning applications may require more complex interactions. For example, mapping all possible actions to a single button press could be frustrating and unsuitable for a time constrained task. The following subsections present the various interface options available.

2.2.1. Gaze

Gaze-Directed interfaces leverage on-board sensors in order to allow users to execute actions based on the direction of their gaze [43]. For example, if a user looks at an object, it is selected either instantly or after a predefined time. This approach has no monetary cost as it makes use of the smartphone's Inertial Measurement Unit (IMU). In testing done by Powell et al. [36], this approach appeared to be more immersive than an external controller in an application where the interface being tested was used to control navigation. However, Powell et al. also found that a gaze-directed approach did not give users as great a sense of control as the external controller. This apparent contradiction between immersion and a sense of control is likely explained by the interface being physically engaging (i.e. tracks a user's natural head movement vs. an abstracted button press) while having a lower degree of expressivity (e.g., users cannot look in a direction without moving in that direction). This issue of low expressivity may also explain the adoption rates of this technique in a survey of VR applications on the Google Play Store [53], where the non-timed version represented 6% of the apps surveyed, while the timed version (called Dwelled Gaze) represented 23%.

2.2.2. Tilt

This category of interfaces refers to a type of gesture recognition which makes use of internal smartphone sensors. This approach allows users to signal actions by moving their VR headset in an uncommon way (i.e. a gesture) [48, 49, 53]. For example, a user can undo an action by tilting their headset sideways. Much like the gaze-directed interface, this approach has no monetary cost as it makes use of the smartphone's IMU. This approach was seen as the third most popular (20% adoption) interface in a survey of VR applications on the Google Play Store [53] and this is likely due to it leveraging hardware in the smartphone, while also avoiding accidental actions by exploiting uncommon movements. This results in a more expressive interface than the gaze-directed approach, while avoiding the cost of additional hardware. Lastly, this type of interface was also found to be more immersive [49] and intuitive [48] than using an external controller in applications related to navigation. This was likely due to it being physically engaging (as with the gaze-directed approach), while also allowing users to look around without initiating navigation.

2.2.3. Computer Vision

In this approach, users use specially designed images called markers to interact with the virtual world [54]. For example, a user can inspect a virtual 3D object related to a physical printed 2D image by rotating the physical image. This is possible because the orientation and position of the virtual 3D object is mapped to that of the printed image using computer vision techniques. This approach leverages the smartphone's processor and camera and thus only has a monetary cost associated with the printing of markers, which is on the magnitude of cents. However, this approach does have a number of challenges associated with it, specifically higher detection latency due to the smartphone camera's shutter speed, higher power consumption due to image processing and lower robustness as image recognition quality is dependent on factors such as lighting and view angles [44]. Another constraint of this approach is that the physical images being tracked need to remain in the line of sight when they are in use [54]. It is speculated that as a result of these challenges, in addition to the computational cost of rendering VR scenes, this type of interface has seen a far smaller adoption in VR in comparison to other approaches that leverage built-in smartphone hardware. More concretely, no implementations were found in a Google Play Store survey [53].

2.2.4. Magnetic Switch

VR Headsets like the Google Cardboard support the use of a magnetic "switch". This is a small magnetic disc that is housed on the side of the VR headset near the smartphone's magnetometer. A user can carry out an action by pulling down on the magnet which moves the magnet closer to the smartphone's magnetometer [44]. For example, a user can signal that they are ready for the next game level by sliding a magnet along the side of a VR headset. While this approach leverages the built-in magnetometer, it does require that a user purchases a magnet that costs on the magnitude of single dollars. This approach has two drawbacks to note. First, not all smartphones have their magnetometer in the same location, meaning that specific phone models may be required for an application. Second, since the magnetometer cannot easily distinguish between different magnets, applications will only be able to make use of one of these magnetic switches and this may limit the degree of possible expressivity. That being said, this approach was seen to be the most adopted interface in a survey of Google Play Store VR apps [53], with an adoption rate of 45%. It is speculated that this approach had a high adoption rate as it allowed users to look around scenes without making accidental selections, a problem prevalent with gaze-directed selections and gesture (i.e. tilt) based selections.

2.2.5. External Electronics

The last type of interface category makes use of external electronics over built-in smartphone sensors. The theme of this approach is that users may achieve more expressivity at a higher monetary cost. Examples of these types of interfaces include conventional external controllers with buttons and 3 axis tracking, specialized touch sensitive surfaces [15, 20, 51] and hand gesture recognizers [27]. This interface approach was amongst the least adopted in a survey of VR applications on the Google Play Store [53], with an adoption rate of 6%. It is speculated that this low adoption may be due to the cost and availability of external controllers. It is believed that this interface choice is too costly for the mobile VR context [44, 49] as the cost is on the magnitude of tens to hundreds of dollars.

Table 2.a. Overview of smartphone-driven VR interfaces.

Type	Components Used	Expressivity	Cost	Adoption Rate [53]
Gaze	Built-in IMU	<u>Low</u> Only allows for selection and selection either happens immediately or after a delay. This interferes with the viewing of a VR scene.	<u>Free</u> Uses built-in sensors only.	5.7% Instant Gaze, 22.8% delayed gaze.
Tilt	Built-in IMU	<u>Medium</u> Only allows for selection, but does not interfere with the viewing of a VR scene.	<u>Free</u> Uses built-in sensors only.	20%
Computer Vision	Built-in Camera	<u>High</u> Allows for 6-axis manipulation of a 3D object via the tracking of a 2D marker.	<u>Low</u> Magnitude of tens of Cents as it uses built-in sensors and printed items.	0%
Magnetic Switch	Built-in Magnetometer and external magnet	<u>Medium</u> Only allows for selection, but does not interfere with the viewing of a VR scene.	<u>Medium</u> Magnitude of single dollars as it uses built-in sensors and a magnet.	45.7%
External Electronics	External electronic components	<u>High</u> Can support 3-axis tracking (e.g., Samsung Gear VR Controller), gesture tracking (e.g., Leap Motion),...	<u>High</u> Magnitude of tens to hundreds of dollars as it uses specialised / dedicated external electronics	5.7%

The findings of the previous subsections (2.2.1 – 2.2.5) have been summarised in table 2.a. and it can be seen that various interfaces solutions have been widely explored in the context of mobile VR, with the exception of computer vision (CV) based solutions. Zikas et al. [54] explore computer vision tracking (feature matching based) in the context of mobile VR, but this work is more focused on enabling 6 degrees of freedom tracking, as mobile VR headsets are currently limited to 3 degrees of freedom. Additionally, no quantitative user testing (e.g., immersion and usability testing) was done to evaluate this solution.

Kroes and Ament [22] also explore a computer-vision based interface, but their interface makes use of LEDs to create trackable features. Their solution is also for desktop-driven VR, though they aim to reduce cost through the simplification of the headset tracking process (i.e., no expensive base stations are needed). This suggests that further research should be done into the topic of computer-vision based VR interfaces, especially given that this approach promises the same high degree of expressivity as external electronics at a lower cost. This cost to benefit ratio is especially useful in the educational context where solutions must scale to classroom sizes. The following chapter details a proposed CV interface for use in educational VR games.

Chapter 3:

System Overview

The literature reviewed suggests that there is merit in researching computer-vision based virtual reality (VR) interfaces due to their ability to provide a high degree of expressivity at a lower relative monetary cost compared to external electronic controllers.

This project thus aims to enable users to control 3D objects in a VR environment through the physical manipulation of real-world image markers. This is achieved by using a camera and computer vision algorithms to track image markers (defined further in section 4.1.) in the real-world. This process produces the orientation and position of markers in world space and this data can be mapped to 3D objects in a virtual environment. This project's proposed solution comprises of two systems and this chapter provides a high-level overview of them. The first system is a front-end Application System that manages hardware interfaces and encapsulates the VR interface and its application. The second system is a back-end Computer Vision (CV) system that is used to track 2D markers in the real-world via the smartphone's camera. These systems work together to enable a user to carry out actions in a virtual environment.

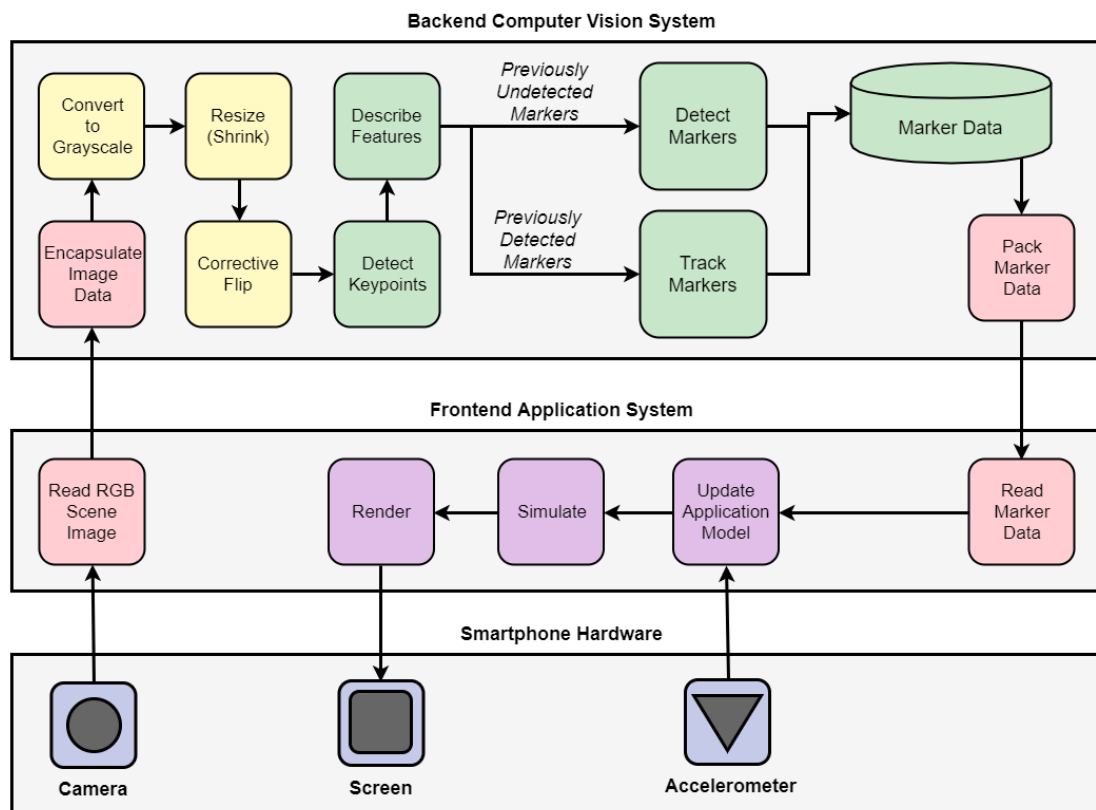


Figure 3.a. A diagram showing the main activities of the two systems needed for marker tracking using a camera. ■ represents activities which manage communication between the two systems. ■ represents image data preparation activities and ■ represents computer vision processes. ■ represents application specific activities and ■ represents hardware components.

3.1. Introduction

In the context of mobile VR, the primary interface of the smartphone (i.e., the screen) is made inaccessible by the VR headset [44]. This project aims to remedy this issue by leveraging printed image markers and the built-in camera. The final result is that users can control 3D objects in a VR environment through the physical manipulation of real-world image markers.

The proposed solution has been separated into two systems, namely the front-end application system and the back-end computer vision system. The application system manages hardware interfaces and contains the VR interface and its application. The computer vision system takes raw camera data provided by the application system and uses it to detect and track markers in view of the camera. The computer vision system then packs the marker information for use by the application system. The high-level design and motivations for these two systems and their interactions are provided in the following subsections.

3.2. System Design

The VR interface solution proposed in this thesis is implemented as two systems. Firstly, there is a system which uses computer vision techniques to identify predefined image markers in a provided image scene. This can be seen as a generalized computer vision back-end system. The second system manages hardware interfaces and encapsulates the virtual reality interface and its application. The second system can therefore be seen as the front-end application system. The overall solution can be viewed as an image tracking pipeline, which starts with the capturing of an image and ends with the rendering of the resulting scene. The flow of activities in and between these two systems is summarized in Figure 3.a. and is further discussed in the following two subsections.

3.2.1 Frontend Application System

The front-end application system communicates with hardware and contains the virtual reality interface and its application. This system is implemented using C# and the Unity3D game engine for convenience. Specifically, Unity3D provides the following benefits:

- The engine provides convenient hardware interfaces. These interfaces allow for pointer-level memory management as well as the ability to read the camera sensor data.
- Unity3D contains a package that automatically manages all aspects of the mobile VR experience. This includes stereoscopic rendering as well as head tracking via the accelerometer.
- Unity3D also supports native libraries. This enables the use of OpenCV and all its computer vision algorithms.

The overall solution proposed in this work can be seen as an image tracking pipeline. In this view, the front-end application system represents the start and end of the pipeline. In other words, these parts of the pipeline are used for the acquisition of raw data and the application of the then processed data.

Inter-System Communication - The application system starts the marker tracking process by reading raw data from the camera and storing it in memory. This information is then made accessible to the back-end computer vision system via a pointer. The computer vision system then uses this raw data to determine the position of markers visible to the camera. This positional data is then packed into a compact format and made available to the application system via another pointer. These activities are represented by the pink elements in Figure 3.a.

VR Interface and Application - Another relevant aspect of the application system is the VR interface and its application. These components are represented by the purple elements in Figure 3.a. The VR application aspect of this system represents the 3D virtual environment that a user is interacting with. The application uses the state, position and orientation of markers to update aspects of the environment. This could take the form of moving a 3D pen object to match the motion of a marker or changing the orientation of a 3D building to reflect the new orientation of the related marker. Additionally, the VR interface aspect of this system represents the means by which a user interacts with the 3D environment. The interface serves two main purposes:

- It provides feedback for a user's actions.
- It serves as a buffer between the information provided by the computer vision system and the virtual 3D world this information influences.

Interface Feedback - When a user manipulates a marker in the real world, it should have consequences for the related object in the virtual 3D environment. These consequences may take the form of positional or orientation changes of virtual objects as a result of positional or orientation changes of markers in view of the camera. Image tracking is also susceptible to reliability issues due to factors like lighting quality and line of sight. It may therefore be important to communicate a marker's visibility to the user so that they can adjust their behaviour for better marker detection. Within this context, the VR interface aims to communicate a marker's state.

Buffering Change - Abrupt changes in marker position, orientation and state may disrupt a user's immersion. This may be caused by the momentary loss of detection of a marker or a momentary change in lighting conditions. Within this context, the VR interface could mitigate disruption by acting as a buffer between the raw marker data provided by the computer vision system and the application system's 3D environment. This is done through the use of interpolation and thresholds to smooth raw marker data in order to reduce abrupt changes in the virtual world. Specifics on how the VR interface has been implemented are discussed further in chapter 5.

3.2.2. Backend Computer Vision System

The backend computer vision system takes a camera image provided by the application system and uses computer vision algorithms to detect and track markers present in the provided image. The state, position and orientation of predefined markers are then packed into memory and made available to the application system for use in the virtual environment. The computer vision system is implemented using C++ and the OpenCV library.

Using OpenCV has the following benefits:

- OpenCV has a library of modern computer vision techniques which can be used for both image detection and image tracking.
- OpenCV is implemented in C++ which makes it compatible with Unity3D via the native coding API.

In the pipeline metaphor mentioned above, the computer vision system represents the middle of the pipeline. In other words, it transforms raw camera data into useful marker information for use by the application system. This part of the pipeline begins by encapsulating the raw camera data in an OpenCV matrix in order to standardize its format.

Data Preparation - Once the data has been encapsulated and is in a format that OpenCV supports, it goes through a series of data preparation steps. These steps are represented by the yellow elements in Figure 3.a. The first step is to change the image from a Red-Green-Blue (RGB) representation to a greyscale representation. This reduces the dimensionality from 3 channels to 1 channel as is required by OpenCV. The image is then resized such that the width and height of the image is proportionally reduced and this further decreases the amount of data that needs to be processed. These reductions aid in increasing the real-time performance of the pipeline. Lastly, the image may be flipped horizontally if the camera provides a mirror image. This is necessary because the computer vision algorithms used in this project make use of relative key-point positioning and a flipped image would have a different key-point representation.

Computer Vision Algorithms - After the data preparation phase, a set of computer vision algorithms are used to determine the state, position and orientation of predefined markers for use in the application system. This segment of the pipeline is represented by the green elements in Figure 3.a. Firstly, a key-point detection algorithm is used to select pixels that may represent distinguishing features in a given image. These features may take the form of something like an edge, depending on the approach used. These pixel keypoints are then further enriched through a feature description process, which adds additional information to the original list of pixels. This enriched data may for instance include surrounding luminosity and it is used to help distinguish pixel key-points from one another. These features are then compared to sets of features which represent different predefined markers and if a number of matches are found, then the corresponding marker has been detected. Once a marker has been detected it may be tracked in future iterations as tracking has been found to have a lower computational cost than standalone detection. Specifics of the image detection and image tracking implementation will be further discussed in chapter 4.

Chapter 4:

Backend Computer Vision System

4.1. Introduction

The aim of the backend computer vision system is to enable the localization of 3D props in the real world. This can be achieved in a simplified manner by attaching a known 2D surface onto a 3D object so that its orientation can be determined relative to a view point. This 2D surface is called a tracker or a marker and examples are shown in Figure 4.a.

For a tracker to be useful, it must be identifiable in a given scene so that its orientation information can be linked to a related 3D object. These trackers can be identified in a bottom-up approach where emerging features are matched to a known collection of features [39]. In this approach, a tracker can be any image and an increasing amount of contrasting points improve its tracking fidelity (see image B and C in Figure 4.a). Trackers, called Fiducial Markers in this case, can also be identified in a top-down approach where encoded data is embedded in the tracking surface [13] (as demonstrated in image A of Figure 4.a.). In this approach, a tracker is specially designed so that decoding its surface provides information that includes an identifier.

The main difference in these two approaches is that the bottom-up approach, called feature matching, gains robustness at the cost of speed. Specifically, a tracker can be identified and positioned with relatively few feature points, while the fiducial markers used in the top-down approach cannot be identified if there is significant occlusion of the encoded data [13]. Fiducial markers have a speed benefit because once a marker's data is decoded, it can be immediately identified. In contrast, feature matching algorithms used in the bottom-up approach require a matching step to link scene pixels with collections of pixels (representing known trackers) and this step can be time consuming even though partial matches are sufficient for identification. Given that this project aims to produce an interface that will be held by users, robustness to occlusion is preferred. For this reason, the remainder of the chapter details the development of a feature-matching based computer vision system.

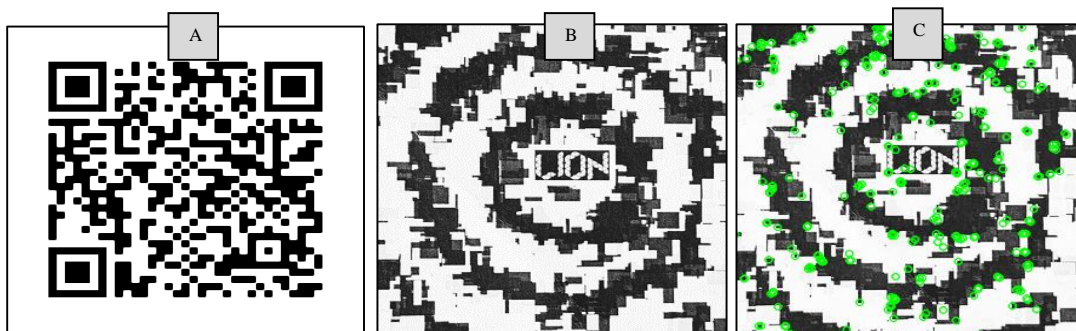


Figure 4.a. These are examples of trackable 2D surfaces. On the left is a QR Code and in the middle and on the right are feature rich images generated by this project. Image C demonstrates points of Image B that were determined to be useful for identifying the tracker by a feature matching algorithm.

4.2. Related Work

Computer vision techniques that make use of feature matching are useful for localizing real world objects in Augmented Reality (AR) and Virtual Reality (VR) [47]. These techniques work by first detecting pixels of interest, formally called keypoints, in a given scene image, based usually on contrast (compared to neighbouring pixels). Another algorithm is then used to generate descriptions of the areas surrounding each keypoint so that each keypoint can be uniquely identified. The combination of a keypoint and its description is called a feature and a set of features can be used to uniquely describe an image. A tracker can be detected in a scene image if enough of the tracker's features are found in the scene. In addition to detection and description algorithms, feature tracking algorithms are also useful when implementing a feature matching system. These algorithms save computation by leveraging previous detections in order to narrow the search space during feature matching, specifically when motion across frames is coherent. The following subsections present algorithm choices for the implementation of a feature matching based computer vision system.

4.2.1. Key Point Detection Algorithms

The first step in a feature matching pipeline is to detect pixels of interest, specifically pixels that can be used to discriminate images from one another. There are a variety of keypoint detection algorithms and this section focuses on three popular algorithms that are included in the OpenCV library. The first is SIFT (Scale Invariant Feature Transform) [23, 24]. This algorithm includes both a detection and description phase and is the oldest of the three approaches. Next is SURF (Speeded-Up Robustness Features) [2] which was created in an attempt to improve on SIFT's relatively slow runtime. These two algorithms are patented and as a result ORB (Orientated FAST and Rotated BRIEF) [39] emerged as a more efficient and non-patented alternative. While efficiency is an important characteristic, robustness is also of value when developing an interface object that is tracked in real time with computer vision. Unfortunately, efficiency and robustness are competing factors as improving one can result in the reduction of the other. The remainder of this subsection will discuss the way in which these three algorithms detect keypoints.

SIFT – The Scale Invariant Feature Transform (SIFT) algorithm aims to detect highly distinctive features that are invariant to small translations and differing scales [23]. This approach begins by producing a pyramid of images such that each level of the pyramid represents the original image at different Gaussian filter strengths, at a specific scale. The Gaussian filtered images at each scale level (called octaves) are then differenced (i.e. approximation of a Laplacian of Gaussians) to produce a Difference of Gaussians (DoG), which acts as an edge detector. Then each pixel in the DoG is checked to see if it is a local extrema. This is done by comparing its value to the surrounding pixels in the DoG using a 3x3x3 volume, such that the third dimension of this volume spans across scale levels. This technique produces a robust keypoint detector as only stable keypoints are consistent across different scale levels and Gaussian filter sizes. Lastly, it should be noted that SIFT is patented and has licensing fees related to it.

SURF – The Speeded-Up Robustness Features (SURF) algorithm was introduced by Bay et al. [2] in an attempt to offer improved performance over SIFT. One of the slower aspects of SIFT is the use of a Difference of Gaussians as an approximation of a Laplacian of Gaussians for use in edge detection. SURF aims to be more efficient by using a box filter to approximate a Laplacian of Gaussians instead. The main benefit of this is that filtering operations allow for parallel processing, meaning that SURF can detect keypoints more quickly. SURF also makes these calculations simpler through the use of summed-area tables. This algorithm is also patented.

ORB – The Orientated FAST and Rotated BRIEF (ORB) algorithm was created by Rublee et al. [39] as a more efficient and non-patented alternative to SIFT and SURF. The keypoint detection phase of this algorithm is based on the Features from Accelerated and Segment Test (FAST) [38]. FAST works by checking a ring of 16 pixels around a given pixel P, and if N contiguous pixels in this ring are brighter or darker than P, then P is a keypoint. This approach is highly parallel and computationally inexpensive when compared to SIFT, though it is also far less robust as it is not robust against scale changes. ORB attempts to remedy the issue of robustness by using a pyramid of images in order to find features that are consistent across multiple scales. These results are then further refined through the use of a Harrison Corner Measure [16] as only the top M ranked keypoints are used.

4.2.2. Key Point Description Algorithms

Once a pixel has been identified as a keypoint, the neighbourhood of pixels surrounding it must be described in order for the keypoint to be detected in different contexts. This descriptive data can be used to uniquely identify a keypoint and when paired with its pixel is called an image feature. SIFT, SURF and ORB contain a description step that is used to produce features and the remainder of this subsection will discuss how this is achieved.

SIFT – SIFT [24] uses a histogram of orientations to describe a given keypoint. These histograms are produced by examining the 16x16 pixels surrounding a pixel and calculating the gradient for each of these. A gradient can represent one of eight cardinal directions. These gradients are then binned to form a 4x4 descriptor such that each bin contains a histogram made up of 16 corresponding gradients. Orientations closer to the keypoint are weighted higher and bins are normalized to unit length. This type of keypoint description ensures rotational invariance and robustness against illumination changes. One issue with this descriptor is that it has a large memory footprint as it contains 128 floating point elements.

SURF – SURF [2] attempts to remedy the performance issue of SIFT by considering wavelet responses in the horizontal and vertical directions. These orientations are produced by examining pixels surrounding a keypoint in a 6x6 grid. The orientation of a keypoint is then determined by using a sliding window of 60 degrees such that it contains as many of the neighbouring pixel orientations as possible. This approach ensures that SURF contains some rotational invariance, while also having a smaller memory footprint (256 bytes) than SIFT (512 bytes).

ORB – ORB's [39] description phase is based on the BRIEF (Binary Robust Independent Elementary Features) algorithm [7]. BRIEF works by creating a binary string that represents the surrounding SxS (default for S in OpenCV is 16) pixel intensities (called a patch) of a given keypoint. These intensities are pre-smoothed using a Gaussian filter to ensure robustness against differing lighting conditions. If a given neighbouring pixel is darker than a keypoint, then its corresponding bit value in the bit string is 1, and if it is lighter it is represented with a 0. This results in a 128-bit descriptor that has a lower memory footprint in comparison to SIFT. ORB adds rotational invariance to this algorithm by orientating each SxS patch to a direction calculated using a corner and the patch's intensity centroid. This approach was seen to be successful as ORB is more than 4 times faster than SIFT and SURF [8].

4.2.3. Tracking Algorithms

The algorithms discussed in the previous subsections are sufficient for localising a tracker in a given scene. One issue with using these algorithms alone is that each time a scene image is received, a detection and description algorithm must be run on the scene before a tracker can be matched and localised. These are expensive operations, as each pixel of the scene will be examined in some manner. A solution to this issue is to use a tracking algorithm that leverages past detections in order to constrain the search space. This works by using previously detected features and information about scene differences to produce updated features. For this project, Mean-Shift [9], Cam-Shift [5] and Lucas Kanade Optical Flow [25] are reviewed as they are included in OpenCV. The remainder of this subsection will discuss how these algorithms work.

Mean-Shift – Mean-Shift [9] attempts to find the new location of a feature by using a histogram back-projection [46] and the location of the previously detected feature. A histogram back-projection is a single channel image with the same dimensions as the scene image such that each pixel represents the probability of that pixel being the feature keypoint. Mean-Shift iteratively moves a window near the previously detected feature location until it contains a certain number of high probability points. This window is moved by continuously calculating the centroid of the contained points and comparing it to the centre of the window. The x and y translations of the window can then be used to find the updated feature point in screen space. One drawback of this approach is that the window has a fixed size meaning that this approach is not scale invariant.

Cam-Shift – Cam-Shift [5] was created to add scale invariance to Mean-Shift. Cam-Shift first uses Mean-Shift to find a candidate centroid, given a histogram back-projection and a previously detected feature point. Once Mean-Shift converges, Cam-Shift updates the size of the window to represent a best fit ellipse. This is then repeated until convergence occurs. The window adjustment provides some scale invariance. One potential issue with Cam-Shift and Mean-Shift is that they are based on histograms, meaning that lighting conditions may affect their accuracy.

Lucas Kanade Optical Flow – The Lucas-Kanade Method [25] differs from the previous two methods in that it examines the pixels around a previously detected feature point, rather than the whole image (i.e. histogram back-projection is of the whole scene image). Specifically, it calculates an optical flow vector using the 3x3 patch of pixels surrounding a feature point via a least square fit method. This produces a vector that describes the motion of a feature point across two scene frames. This process is also applied to the scene image at different scales in order to find both small and large optical motions. This makes the method more scale invariant. Additionally, because this method makes use of intensities instead of histograms, it is also more robust against changes in scene lighting and ambiguity caused by multiple trackers being in view.

4.3. Implementation Details

A feature matching pipeline was implemented that supports SIFT, SURF and ORB. This was done to allow for testing and comparison across approaches. Since this pipeline is being used in the VR context, where framerate is important, the pipeline was also built to support image tracking via the Lucas-Kanade Method for comparison purposes. The other two image tracking methods were not supported due to their comparatively lower robustness under scale changes, lighting condition changes and colour ambiguity caused by multiple in-scene trackers. This section details the algorithmic structure of the pipeline along with the development of a marker generator.

4.3.1. Marker Generator

This project makes use of feature matching, a computer vision technique that distinguishes images based on feature points using SIFT, SURF and ORB. These three techniques find feature points based on intensity which essentially implies high contrast between adjacent points. In order to facilitate the development of this pipeline and the overall interface, a marker generator was developed. This generator was written in Python 3 and can produce markers with high visual contrast based on a set of supplied parameters. The parameters are the width, height, random seed, border width, texture style, text and scale of the texture, and examples of available styles are included in figure 4.c.

The tool uses a procedural approach to generate markers. Specifically, it places random grayscale rectangles using equations based on the texture style as shown in table 4.b. While these trackers may bear a resemblance to fiducial markers such as QR codes, these markers do not encode any information. Additionally, the borders around the edges of the trackers serve aesthetic purposes only and do not aid in detection unlike in the case of fiducial markers.

Given that the final solution of this project will only make use of a relatively small number of markers (i.e. less than 10), it was not necessary to evaluate the quality of these markers in terms of feature overlaps, and the randomness introduced by the placement algorithm proved sufficient for ensuring uniqueness. For solutions with a large number of trackers, it would be necessary to ensure that each marker contains enough unique features. Each marker was run through the final pipeline to ensure that it was correctly identified.

Table 4.b. This table contains pseudo code for the Marker Generator for placing rectangles according to each of the tracker styles. Examples of the result are provided in Figure 4.c.

Style	Placement Algorithm
Radial	<pre>x = random(0, width); y = random(0, height); colour = random(0, 255); radial = power(power(x-width/2,2) + pow(y-height/2,2),0.5); if (radial + random(jitter) % threshold > threshold / 2) place_rect(x, y, colour);</pre>
Dots	<pre>x = random(0, width); y = random(0, height); colour = random(0, 255); if (((x + jitter) % threshold > threshold * 0.2) and ((x + jitter) % threshold < threshold * 0.8) and ((y + jitter) % threshold > threshold * 0.2) and ((y + jitter) % threshold < threshold * 0.8)) place_rect(x, y, colour);</pre>
Stripes	<pre>x = random(0, width); y = random(0, height); colour = random(0, 255); stripe = (x + jitter) + (y + jitter) if value % (threshold) > (threshold * 0.5): place_rect(x, y, colour);</pre>
Random	<pre>x = random(0, width); y = random(0, height); colour = random(0, 255); if random(0, 1) > 0.5: place_rect(x, y, colour);</pre>



Figure 4.c. Various marker styles that the generator tool can produce. In order from left to right the styles are Radial, Dots, Stripes and Random.

4.3.2. Image Detection Process

The image detection and description portion of the pipeline was implemented using the best practices described in the OpenCV documentation. The process can be summarized as follows:

1. *Initialise*: When the library is launched, it reads in registered marker images and builds a dataset of image features for each marker.
2. *Process Scene*: The pipeline is fed in a scene image. This is captured from a device's camera or from a local video file.
 - a. *Load Scene*: The scene image is read in from a pointer and converted to an OpenCV friendly format.
 - b. *Scale Scene*: If enabled in the pipeline settings, the scene is scaled down to reduce the amount of data being processed.
 - c. *Scene Detect*: A detection algorithm is run on the scene image to extract keypoints.
 - d. *Scene Describe*: A description algorithm is run on scene keypoints so that they can be matched to known tracker features at a later stage.
3. *Scene Match*: The scene's feature points are matched to each of the known trackers using the K-nearest neighbours (KNN) algorithm ($k=2$).
 - a. *Hamming Brute Force Matching*: For ORB, the Hamming Brute Force matcher is used as it is more suited to finding the distance between bit string descriptors.
 - b. *Flann Matching*: The FLANN based matcher is used for SURF and SIFT as it is better for finding the distances between floating-point based vector descriptors.
 - c. *Ratio Test*: For each of the matched tracker features, there are two matched scene feature points due to the use of KNN ($k=2$). The first point of the set is the closest distance-wise and the second is the second closest distance-wise. These two points are compared using a ratio test (threshold=0.9) to ensure that the first closest match is a significantly closer match. This process filters out ambiguous matches.
4. *Homography Calculation*: If more than 20^4 matched features are found then a homography representing the transformation between the tracker and the camera view is calculated. The Random sample consensus algorithm (RANSAC) [14] is used to remove outliers that disagree with the homography. The translation and rotation of the tracker is then extracted from the homography through its decomposition.

⁴ This is 5 times the minimum required amount (i.e. 1 for each corner). This prevents filtered-out false matches from degrading the homography. It also makes tracking (see section 4.3.3.) more robust against features lost over time.

4.3.3. Image Tracking Extension

The feature matching pipeline was also extended to enable tracking. This was done by attaching a state to each tracker. If a tracker was already identified in a previous scene image, then it is not necessary to detect and describe the scene. Rather the Lucas-Kanade method can be applied using the previous scene image, the previously matched keypoints of the tracker and the new scene image. This process is represented as follows: (continuing numbering from section 4.3.2):

2. *Process Scene*: The pipeline is fed in a scene image. This is captured from the device camera or from a local video file.
 - a. *Load Scene*: The scene image is read in from a pointer.
 - b. *Scale Scene*: If enabled in the pipeline settings, the scene is scaled down to reduce the amount of data being processed.-- If Marker has state 'Not-Found' --
 - c. *Scene Detect*: A detection algorithm is run on the scene image to extract keypoints.
 - d. *Scene Describe*: A description algorithm is run on scene keypoints so that they can be matched to known tracker features at a later stage.-- If Marker has state 'Detected' or 'Tracking' --
 - e. *Feature Tracking*: The previous scene, previous scene's matched feature points and new scene are processed using the Lucas-Kanade Optical Flow method. This generates the feature points needed for the next matching step. If not enough features are tracked (i.e. lost) then the tracker's state is updated to 'Not-Found'.

4.4. Evaluating the Pipeline

4.4.1. Performance of the Image Detection Pipeline

In order to continue the project's development, SIFT, SURF and ORB must be compared to see which is most suitable for a computer-vision based interface. This assessment aims to compare the algorithms both in terms of accuracy and speed. This evaluation was run on a laptop with an i7-4710HQ CPU (2.49 GHz, 4 Core, 8 Thread) and 16GB of RAM. The library itself was executed using a single thread, meaning that the extra cores did not contribute to the measured performance. The test was run on a laptop instead of a smartphone as extracting performance data and results from a smartphone when using OpenCV is non-trivial. The performance of the library on a smartphone is assessed indirectly in the final evaluation of the interface.

A synthetic evaluation was created to assess SIFT, SURF and ORB. This evaluation works by imposing an image of a tracker onto a moving video using specific translations and rotations (see Figure 4.d). This is useful as accuracy can be assessed on a pixel level by comparing the predicted location of a tracker with its actual location. This would be non-trivial in a non-synthetic evaluation as a sophisticated physical mechanism would likely be needed to accurately measure a tracker's orientation and position. The location of a tracker in this evaluation is specifically the location of each of its corners in screen space. This is useful for this evaluation because the pipeline can only accurately predict the corner locations of a marker in screen space if it accurately determines the homography that explains the translation and rotation needed to get from the camera view point to the tracker.

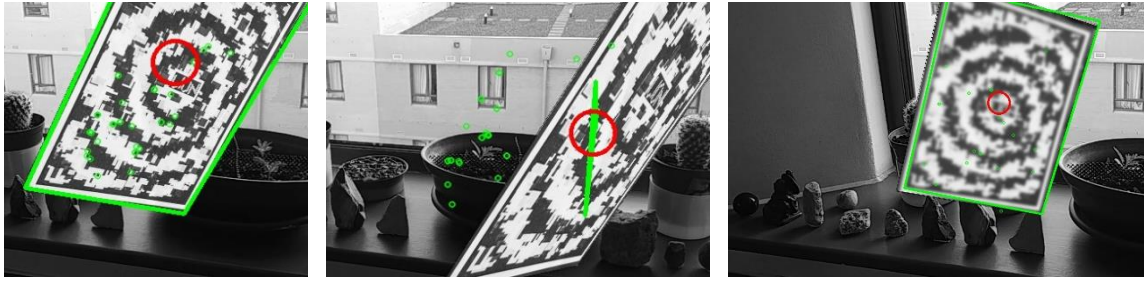


Figure 4.d. Example scene frames from the ORB synthetic evaluation, during which a virtual tracker image was imposed on a pre-recorded video. The left image demonstrates an accurate detection as the green border tightly matches the tracker. The middle image shows a poor detection, likely due to the extreme rotation of the tracker. The right image shows what a normalized pixel error of 1.01 looks like (explained later in this subsection).

The synthetic evaluation comprises 4 tests, each containing 60 scene images with each frame having an interpolated tracker transformation based on the allowed ranges described in Table 4.e. These ranges were chosen experimentally such that the marker is always in view, even under extreme transformations. The first test assesses prediction accuracy of x-axis and y-axis translations in world space using scene units. The second test extends the first test by assessing z-axis translations as a means of testing scale invariance. The third test extends the second by testing rotations in all three axes and this provides a strong indication of rotational invariance. The final test extends the third by assessing robustness against tracker blurriness. This is a useful evaluation, as fast moving trackers may be effected by motion blur, with reduces the clarity of the tracker.

Table 4.e. Synthetic test parameters for the evaluation of SIFT, SURF and ORB. Translation Ranges are given in Scene Units.

Test ID	Frame ID Range	X Translation Range	Y Translation Range	Z Translation Range	X Rotation Range	Y Rotation Range	Z Rotation Range	Gaussian Blur Filter Size
1	1-60	100-500	0-150	1400	0	0	0	0
2	61-120	100-500	0-150	800-2400	0	0	0	0
3	121-180	100-500	0-150	800-2400	-45°-45°	-45°-45°	-45°-45°	0
4	181-240	100-500	0-150	800-2400	-45°-45°	-45°-45°	-45°-45°	1-30

Table 4.f. Percentage of scene frames that were incorrectly predicted for SIFT, SURF and ORB in a synthetic evaluation containing 4 tests with 60 scene frames each.

Test ID	Prediction Accuracy across Scene Frames		
	SIFT	SURF	ORB
1	100%	100%	100%
2	100%	88%	100%
3	88%	53%	82%
4	65%	27%	47%

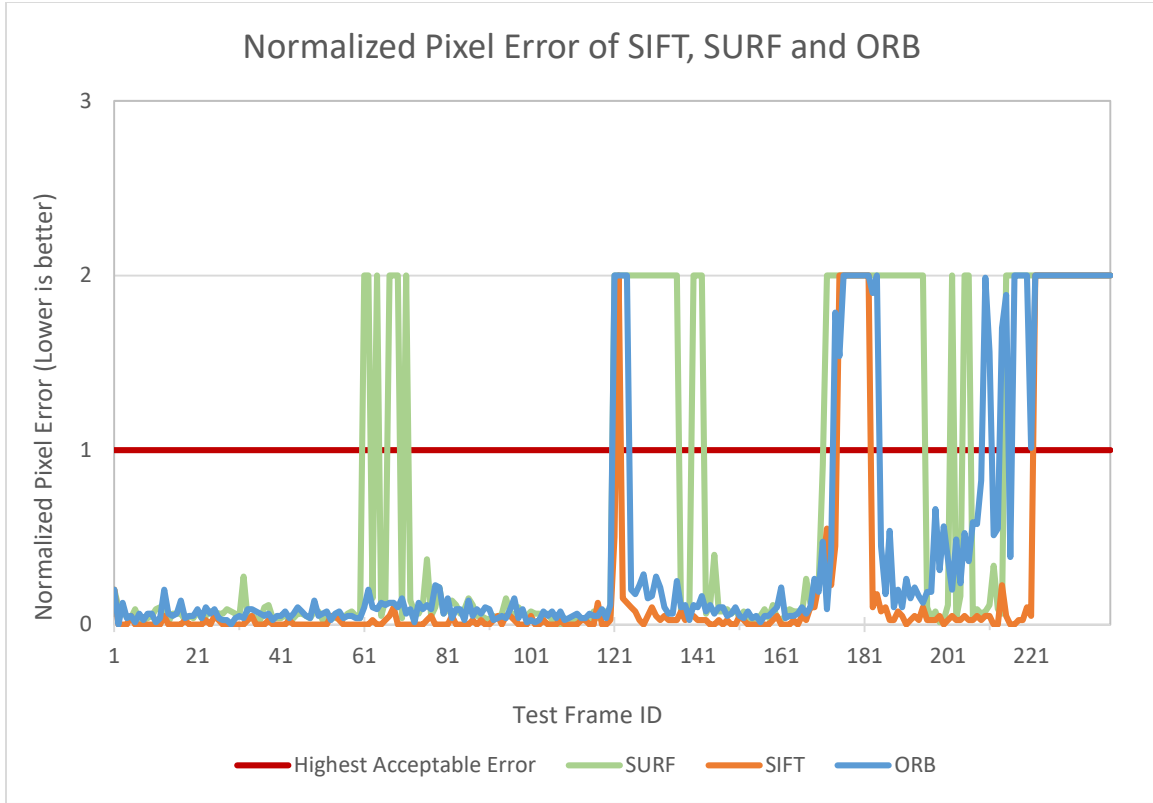


Figure 4.g. The performance of SIFT, SURF and ORB in a synthetic evaluation. A correct prediction must be within 10 pixels for each axis (x, y) for each corner (i.e. 4). The pixel accuracy for each scene frame was normalized by dividing it by 80 and allowing only a maximum value of 2 (for better visualization). A frame is correct if the normalized pixel error is below 1.

Test 1 – As seen in table 4.f, all three algorithms perform well by correctly predicting the location of the tracker in a variety of x and y -axis translations. For this evaluation, each coordinate (i.e. x, y) of each corner is allowed to be incorrect by up to 10 pixels. This error value was determined experimentally as shown in the right image of Figure 4.c. This means that a prediction must be within 80 pixels of the true position to be considered correct. This value was chosen as the correct translation and rotation of a tracker can still be recovered. SIFT can also be seen to be highly accurate in Figure 4.g. as its trend-line remains close to the x -axis origin throughout the test. ORB’s prediction accuracy in contrast fluctuates more, although this fluctuation remains within reason.

Test 2 – This test examines an algorithms scale invariance. As see in table 4.f, SURF was the only algorithm to incorrectly predict scene frames (12% incorrect). These errors specifically occurred when the tracker was further away from the camera view point (i.e. scaled down). This is problematic as users are likely to hold a tracker at different distances from the smartphone’s camera.

Test 3 – The third test aimed to investigate the rotational invariance of the algorithms, by rotating the tracker by up to 45° in all three rotational axes. All three algorithms struggled with prediction accuracy in the case of large rotations and SURF performed significantly worse with almost half of its predictions being incorrect, as seen in table 4.f.

Test 4 – The final test attempted to predict robustness against motion blur by applying a Gaussian blur to the tracker. The Gaussian blur values shown in table 4.e. represent the filter size of the blur operation, meaning that larger values result in more distortion. It should also be noted that this test includes the translations and rotations of previous tests, meaning that previous errors are essentially carried forward. This is a desired effect as these tests are meant to replicate potential real-world scenarios as closely as possible. In this test SIFT performed most optimally, though this performance should only be considered in a relative sense as large amounts of blurriness do not represent the general operating conditions of the system, i.e. blurriness is more of an anomaly.

In general, of the three algorithms tested, SIFT performed the most optimally (88% accuracy) with ORB coming in a close second (82% accuracy) when only considering translations and rotations. When factoring in the timings listed in table 4.h, it can be seen that ORB is a good choice as its detection and description time (68ms) was almost one tenth that of SIFT (554ms). In other words, it is worthwhile trading a 6% loss in accuracy for a 10x speedup, especially considering that this speedup would enable near real-time performance.

Table 4.h. This table contains the time performance of the algorithms being examined, in milliseconds. The numbering in the Algorithm Step column corresponds to that of section 4.3.2.

Algorithm Step	SIFT	SURF	ORB
1. Initialise	255ms	1077ms	928ms
2.c. Scene Detect	292ms	133ms	39ms
2.d. Scene Describe	262ms	291ms	29ms
3. Scene Match	13ms	74ms	1ms
4. Homograph Calculation	13ms	40ms	16ms

4.4.2. Performance of the Extended System

As detailed in the previous subsection, the pipeline's slowest recurring segment is when a scene's keypoints are being detected and its features are being described. This time cost can be reduced through the use of image tracking techniques, specifically using the Lucas-Kanade method [25]. This evaluation was run in a similar manner to that of the previous subsection, but with the image tracking functionality enabled as detailed in section 4.3.3.

Table 4.i. Percentage of scene frames that were incorrectly predicted by ORB (with and without tracking enabled) in a synthetic evaluation containing 4 tests with 60 scene frames each. The accuracy within 160px is also included as Figure 4.k. demonstrate that this is still useful.

Test ID	Prediction Accuracy across Scene Frames		
	ORB	ORB + Tracking (Standard 80px Error)	ORB + Tracking (160px Error)
1	100%	100%	100%
2	100%	88%	98%
3	82%	38%	57%
4	47%	17%	28%

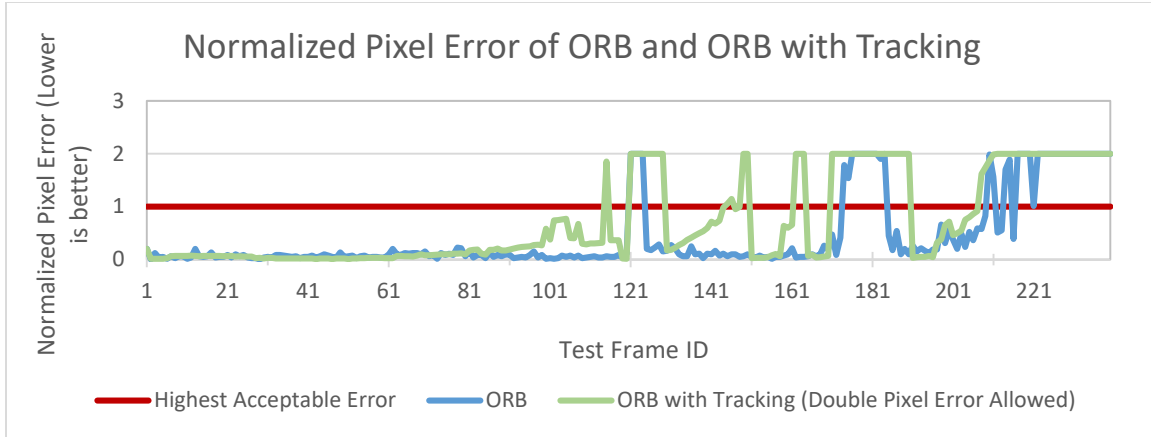


Figure 4.j. The performance of ORB and ORB + Tracking in a synthetic evaluation. A correct prediction must be within 10 pixels for each axis (x, y) for each corner (i.e. 4). The pixel accuracy for each frame for each algorithm was normalized by dividing it by 80 and allowing only a maximum of value of 2. A frame was correct if the normalized pixel error was below 1.

After testing the ORB algorithm with tracking enabled, it was found that the error rate increased significantly when considering the previously used 80 pixels of acceptable error. When examining the results of this evaluation, it was found that a total pixel error of 160 pixels (i.e., double the amount) may also be acceptable as the predicted orientation is still close enough visually to the actual orientation (see Figure 4.k). Therefore, when considering the “double error” rate in table 4.i, it can be seen that tracking reduces the rotational invariance, resulting in a more significant error rate (18% vs 43%). According to Figure 4.j, these rotational errors occur more when rotations are greater than 30° . In terms of timings, ORB without tracking spent 92ms on average on each “Process Scene” step (see section 4.3.3), while ORB with tracking spent on average 56ms (including necessary detection and description). Additionally, the tracking step only takes 11ms versus ORBS detection (39ms) and description (29ms) time of 68ms. This speedup suggests that image tracking is a useful way to speed up the pipeline, especially if the prop being tracked does not require large during use.

This suggests that the final system should make use of ORB with Lucas-Kanade based tracking in order to achieve a balance of speed and accuracy. In addition to this, the final system will make use of scene scaling, which reduces detection (19ms to 6ms), description (29ms to 16ms) and tracking (11ms to 2ms) times by reducing the scene’s dimensions proportionally (width from 720px to 640px). This had a minimal impact on accuracy (Test1: 100%, Test 2: 95%, Test 3: 43%). Lastly, the final system will be configured to use multi-threading to enable the use of multiple trackers and simultaneous detection and tracking as the smartphone being used has multiple cores.

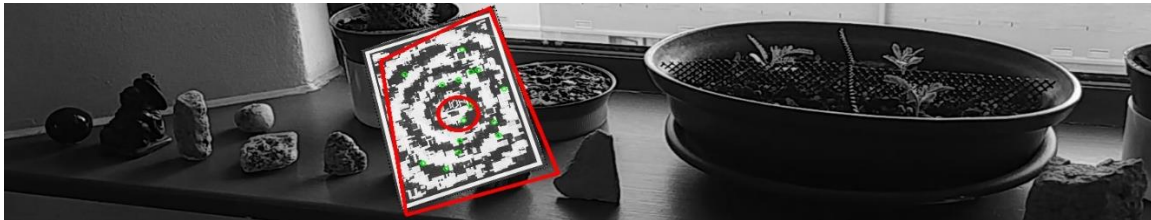


Figure 4.k. This is an example of a prediction error of roughly 160 pixels. The red border represents the predicted orientation, which is visually similar to the actual orientation.

Chapter 5:

Front-end Virtual Reality Interface

One of the aims of virtual reality is to find ways to more naturally interface with virtual environments in order to increase immersion, as shown in the original work by Sutherland [45]. This is useful for educational games as stronger engagement results in better performance by students [42]. While mobile VR experiences are less expensive to implement in terms of monetary cost in comparison to desktop counterparts, it has issues related to interface choice. This is because the main interface (i.e., the touch screen) of the smartphone is obstructed by the VR head-mounted display. This chapter details the research and development of a computer-vision based interface for educational mobile VR experiences. This includes related work, the iterative user-centred design process and the development of an educational environment useful for evaluation.

5.1. Immersive Serious Games

VR allows for a standardized, no-risk learning environment, which is useful in a variety of fields including medicine [35]. No-risk learning means that students can learn about the consequences of actions without damaging property or harming living beings. These types of educational experiences are called *serious games*, as they aim to be both educational and entertaining. Dicheva et al. [12], distinguish serious games from traditional learning activities (e.g., lectures) by pointing out that they have been enhanced with game mechanics. These mechanics may include rapid feedback, visible status (e.g., an always up-to-date score), freedom to fail and social engagement. VR compliments serious games by offering more natural [45] and immersive ways of engaging with educational content. This is useful, as it has been demonstrated [42] that students perform better when their content is more engaging.

Serious games can be evaluated in terms of their entertainment value and learning impact [3]. Specifically, learning impact can be evaluated by comparing the knowledge retention of a control group, taught using traditional learning methods and a group taught using a serious game. The entertainment value would be evaluated by looking at user engagement and immersion. Typically, immersion and user engagement are evaluated using performance metric analysis (e.g., task completion times), interviews and surveys such as the Game Experience Questionnaire [17].

5.2. Iterative Design Methodology

The interface proposed in this project was developed following an iterative User Centred Design process (see figure 1.b in section 1) [34]. In this design process, users are involved at various stages of the development in order to gain insight into the effectiveness of the current interface implementation. Additionally, this process is carried out iteratively in order to further refine the final interface. The design process in this project is implemented in three stages and this process can be seen as the application of the computer vision library developed in earlier chapters.

The design stages used to develop and realise the final interface are as follows:

- **Stage 1: *Defining the Interface.***
The requirements of the proposed interface will be discussed further in the remainder of this subsection.
- **Stage 2: *Exploring a low-fidelity implementation of a potentially viable interface.***
In this stage, a paper prototype was created to test a potential implementation of an interface that met the specifications outlined in stage 1. This prototype also included an educational game for the prototype interface. A paper prototype was deemed ideal for this stage of the design process as it allows for the testing of interface ideas without the need for time-consuming programming.
- **Stage 3: *Exploring a refined high-fidelity implementation of a viable interface.***
This interface must match the requirements of stage 1 and the findings of stage 2. In this stage, the final computer-vision based interface approach was developed and tested in iterations. This includes a physical 3D printed and corresponding virtual software implementation. Furthermore, an educational virtual reality game was developed to make use of this interface and this game will be used in the final evaluation (see chapters 6 & 7).

The first stage of this design process was to define the requirements of an interface that will be effective in the context of a smartphone-driven virtual reality experience. Norman, presents four rules for design in his classic book *The Psychology of Everyday Things* [33]. These rules are summarised as follows:

1. **Affordance:** A user should be able to determine what actions are possible.
2. **Signifiers:** The interface should have visible properties.
3. **Feedback:** A user should be able to determine the current state of the system.
4. **Mapping:** The interface should provide a natural mapping between a user's intentions and the actions required to realise these intentions.

While other design principles exist [31], Norman's four classic rules were chosen for use in this section due to their simplicity. These rules provide a framework that is useful for reasoning about the requirements of an interface that would be useful in the context of an educational virtual reality experience. In accordance with the rules presented above, a useful virtual reality interface should:

1. **Display information for the user.** This adds visible properties to the interface (signifiers) that allows a user to understand the environment that they are in (Affordance).
2. **Allow a user to select a value.** Users must be able to effect the virtual world in some way and this intention is created through the selection of a value. The interface should constrain this choice so that it is in the realm of possibility, for example, not being able to choose index 4 when only 3 choices exist for a multiple choice question (Affordance).
3. **Allow a user to submit a selected value.** This allows a user to effect the state of the educational environment through the submission of a selection, which can be seen as a realisation of an intention (Mapping).

This list of interface requirements can be seen as a list of minimal requirements required to satisfy the design rules presented by Don Norman [33]. The following sections in this chapter will describe in further detail the iterative user centred design process that was carried out and driven by the above list of interface requirements.

5.3. Low-Fidelity Paper Prototype

A paper prototype was designed and developed to test a potential implementation of an interface that matches the requirements listed in the previous subsection. This prototype was inspired by an educational accounting game called Capsim⁵ and in it, students manage an orange stand in a competitive market. In this game, players can buy stock, set a selling price and invest in advertising, which increases the likelihood of sales. This game was chosen as it is conventionally played using spreadsheet templates that lack a visual component. The paper prototype is shown in figure 5.a. and the components of the paper prototype are described in table 5.b.

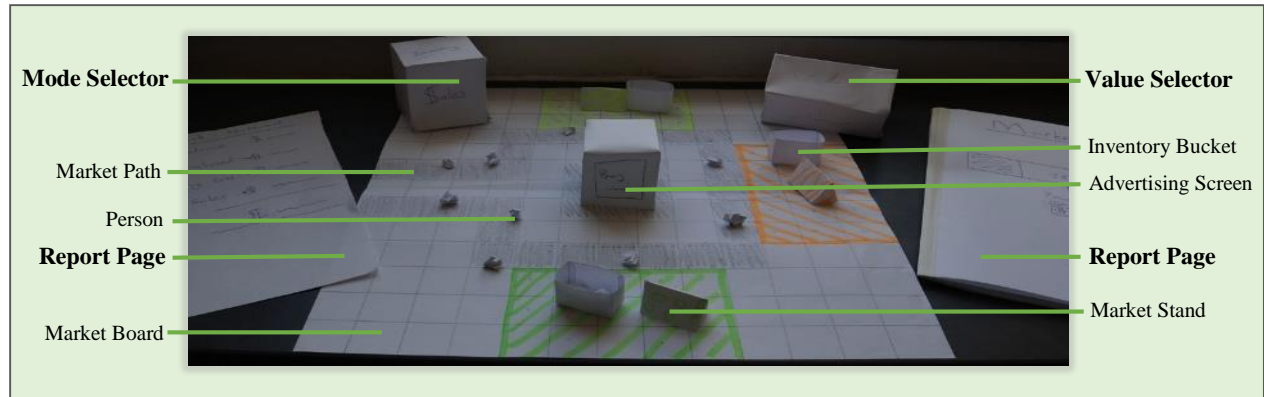


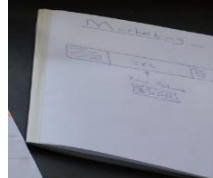


Figure 5.a. A paper prototype of a market simulation game. The interface is comprised of the *Mode Selector*, *Value Selector* and *Report Pages* (i.e. these would have physical implementations). The remaining components (not in bold) are a visual representation of how the game would appear in VR (i.e., purely virtual and not part of the interface)

Table 5.b. Interface components of the paper prototype. *T*

Components	Description
	<u>Mode Selector</u> This cube allows a user to select their interface mode. Each mode specialises the interface for a specific task. The modes are Inventory (for buying stock), Sales (for setting a selling price) and Advertising (for setting an advertising budget). Selection occurs when a user physically rotates the cube and the mode facing upwards is selected.
	<u>Value Selector</u> This prism allows a user to select a value for the current interface mode. This is either a currency for the sales and advertising modes or a number for the stock mode. The user rotates the prism up or down (about the axis that passes through centres of the pentagon faces) to increase or decrease the current value under consideration.
	<u>Report Page</u> There are four report pages. There is a page for each mode and one for the stand's bank statement. Only the pages of the current interface mode and the bank statement are shown at any given time.



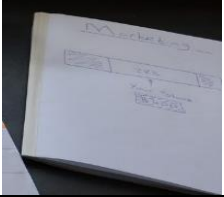
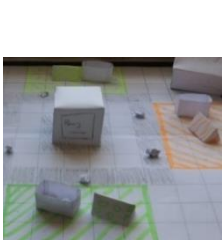
⁵ Available at <https://www.capsim.com/>

The paper prototype described above was evaluated using a group of six computer scientists with backgrounds in Computer Graphics. The game components were manually manipulated, while game calculations were carried out using a spreadsheet template. The game steps were as follows:

1. The current player has at most 2 minutes to execute their turn. They may buy inventory, set their selling price and spend money on advertising. These values are recorded in the spreadsheet template at the end of the turn.
2. Then it is the next player's turn (three players per game). Step 1 is repeated until all players have played their turns sequentially.
3. The spreadsheet is executed and all report page values are updated. The game board visuals are also updated and this includes inventory buckets filling based on remaining stock, people distributing between stalls based on their relative sale volume and the advertising screen playing the advertisement of the player with the biggest advertising budget.
4. Steps 1 to 3 are repeated for 3 rounds and the winner is determined after the final round based on the highest amount of unspent money.

The evaluation consisted of one play-through with 3 of the 6 users choosing an observer role. The feedback of this evaluation is summarised and categorised in table 5.c below.

Table 5.c. Negative (-) and Positive (+) feedback from users categorised by interface type.

Components	Description
	<p><u>Mode Selector</u></p> <ul style="list-style-type: none"> - Shape incorrectly suggests to users that they must roll this selector instead of just placing it. This is an affordance issue and the users proposed that a socket (i.e., motion limiter) could resolve this issue.
	<p><u>Value Selector</u></p> <ul style="list-style-type: none"> - The physical interface does not communicate and enforce a limit. - There is no way to control the size of increments. + A simple interface item that requires no electronics to use.
	<p><u>Report Pages</u></p> <ul style="list-style-type: none"> + Users felt that 4 items were too many parts to hold, especially if a HMD is being used. It may be useful to bundle these into a single book.
	<p><u>Gameplay</u></p> <ul style="list-style-type: none"> - In the paper prototype, the player stalls would increase in height to visualise how much money a player has. This may occlude the view of other players. - The game dynamics may encourage a runaway winner. The player with the most money can advertise more thus increasing sales, which allows for further advertising. This may reduce the enjoyment for other players, especially in an educational setting.

The overall consensus drawn from the feedback in table 5.c was that the complex game rules may take focus away from the interface evaluation. Users also felt that there were too many interface objects, and that a more unified user interface may be easier to learn and use.

5.4. High-Fidelity 3D-Printed Prototype

The user feedback for the paper prototype in the previous section provides insights that are useful for iterating the design of the virtual reality interface being developed. Using these insights along with the requirements outlined in section 5.2, a new high-fidelity interface and testing environment was developed. The rationale behind the new design is described in this section and details regarding each stage of its iteration are given.

5.4.1. Physical Interface Development

The interface developed in this stage of the project has a high-fidelity 3D printed component for use in the educational smartphone-driven virtual reality context. This interface was developed in high fidelity as it is intended for use in the final evaluation of this project. This design was built to fulfil the requirements described in section 5.2, while also being an improvement over the low-fidelity multi-component prototype that was developed and tested in the previous subsection. The new interface design has both a physical and software implementation. The physical part of the design relates to the 3D printed prop that users will hold, while the software relates to the visual and auditory cues used in the virtual reality environment. This subsection will describe the design rationale and development process of the physical part of the interface design.

The following tools were used to develop the physical elements of the interface:

- **Blender v2.80⁶**: This 3D modelling tool was used to develop the interface's mesh-based models, useful for 3D printing and visualization in VR. This tool was chosen for its diverse export options and extensive editing toolset.
- **Cura v4.4.0⁷**: This 3D slicer was used to convert the 3D models produced by blender (STL format) to printing instructions useful for 3D printing (GCode format). Cura was set to use the default printing profile for the Creality Ender-3. Beyond the default settings, an infill of 20% was used to keep the prints lightweight.
- **Creality Ender-3**: This 3D printer was used to carry out the GCode produced by Cura. The printers are housed in a dedicated printing room, ensuring a consistent printing experience. This printer used PLA as a printing material due to its ready availability and print quality.

The first iteration of the high fidelity interface was developed to address the issues outlined in the paper prototype feedback. The new designed is depicted in figure 5.d. and address the following considerations:

1. The interface is a single unified controller instead of multiple separate controllers. This makes the interface easier to handle, given that the VR headset obstructs the user's view of the real world and can make it difficult to pick up and put down physical objects.

⁶ Available at <https://www.blender.org/>

⁷ Available at <https://www.ultimaker.com/software/ultimaker-cura>

2. The interface is shaped like a clipboard to make it easier to hold. The shape also offers greater affordance over the paper prototype as the clipboard shape more naturally ensures that the prop is held correctly with the tracker surface facing the smartphone's camera.
3. The selection mechanism also aims to improve affordance since it is shaped like a gear. This shape provides a tactile feel and more naturally suggests that it should be rotated.

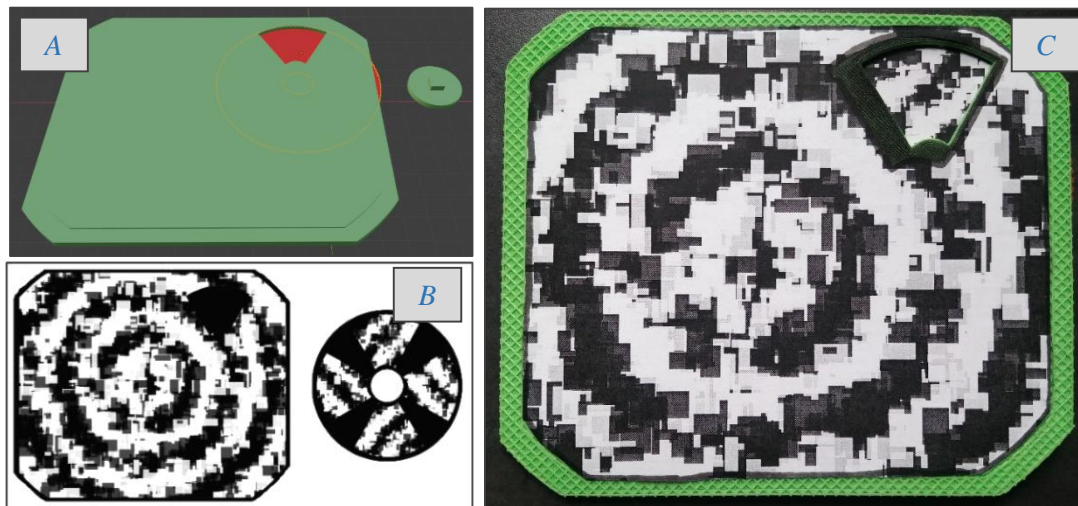


Figure 5.d. Iteration 1 of the High Fidelity Prototype. (A) the 3D models as seen in Blender and (B) the printed tracking surfaces. (C) the assembled 3D-printed interface with the tracking surfaces glued on.

Iteration 1 of the interface is made up of three 3D printed parts (See Figure 5.d (A)). There is a rectangular base, which has a small window on the top right, a toothed gear and a stopper for holding the gear in place. There are also two printed tracking surfaces (Figure 5.d (B)) that enable computer-vision based tracking. These surfaces were generated by the random placement of grayscale rectangles as described in section 4.3.1. The assembled result is shown in figure 5.d (C).

Iteration 1 of the controller was evaluated by a group of experts with a background in Computer Graphics. The following weaknesses were identified during this evaluation:

1. The selection mechanism favours right-handed users as the selection wheel is only accessible from the right-hand side of the base.
2. The selection wheel and window are too small to allow for reliable tracking when held at arm's length (roughly 1m) from the smartphone's camera.

Iteration 2 of the controller was developed to address these issues and is depicted in figure 5.e. This version of the controller features a square base and a larger selection wheel that supports both right and left-handed users. The size increase of the selection wheel and its window also improves visibility of the related tracking surface, which in turn improves tracking reliability at an arm's distance. Iteration 2 also includes a selection guard (marked with a blue X in figure 5.e.) that reduces selection ambiguity as discussed further in this subsection. Lastly, the style of tracker used on the selection wheel was also updated.

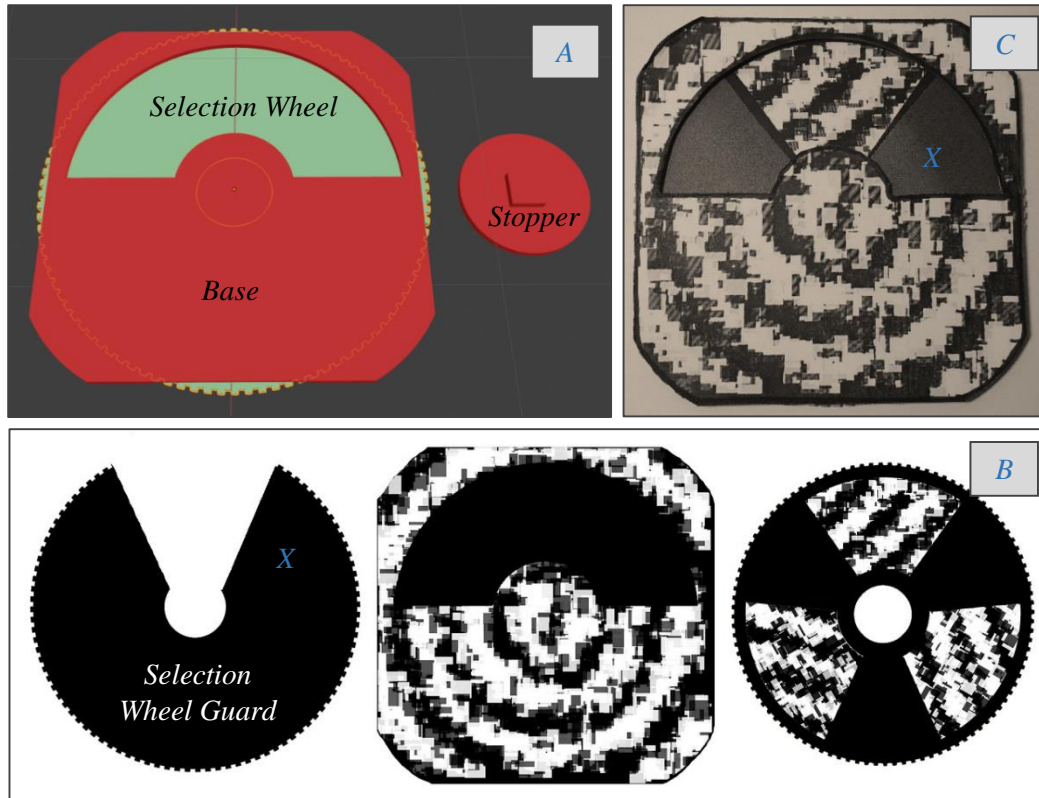







Figure 5.e. Iteration 2 of the High Fidelity Prototype. (A) the 3D models as seen in Blender, and (B) the new printed tracking surfaces. (C) the 3D printed (assembled) interface with the tracking surfaces pasted on.

A number of selection wheel tracking surfaces were explored during the development of iteration 2 of the controller. The surfaces explored are summarised in table 5.f. and these are presented in the order in which they were evaluated. Testing of the various approaches suggests that different designs optimise among the following three characteristics:

- **Degree Level Tracking vs Quantised States:** The tracking approach can either provide the angular rotation of the selection wheel or a quantised detection of when the wheel has been rotated by a specific amount. Angular tracking is useful in animating the virtual representation of the selection wheel, though this approach was outperformed by the quantised states approach in terms of tracking reliability. This is because the later approach prevents extremely rotated feature points from being viewed (improves feature point detection) and allows for most of the features points of a tracker to be viewed when the tracker is aligned with the viewing window. *The most reliable choice is preferred.*
- **Tracking Reliability:** This relates to the ability of the computer vision framework to accurately detect a tracker. This ability improves when a tracker has features that are visible from different distances. *Increased reliability is preferred.*
- **Detections of False Rotations:** This is an issue with the quantised approach where a rotation (state change) is detected incorrectly when feature points of two different trackers are in view. This may result in an oscillation of rotation detections. This may be frustrating for users as they lose the ability to select options. *Reduction in false detections is preferred.*

Table 5.f. The various selection wheel surfaces explored during iteration 2.

Type	Example	Description	Evaluation
A		A single large tracker spread across the selection wheel. The intention is to track the exact orientation of the selection wheel.	<ul style="list-style-type: none"> + Simpler to implement - Unable to reliably track as most (>50%) of feature points are hidden behind the base plate. - Rotated feature points are also more difficult to detect accurately.
B		An iteration over type A where 3 trackers are used with no spacing in-between. This means that all feature points of a tracker are visible when in front of the base plate's window.	<ul style="list-style-type: none"> + Improved tracking accuracy as feature points are never viewed in an extreme rotation. - More difficult to implement as two trackers (current and previous) need to be interpreted in order to determine wheel rotation direction. - Produces false rotations when two trackers are equally in view of the base's window.
C		This design attempts to reduce the false rotation detections produced by type B when the selection wheel is in an intermediate position. The space between trackers is increased to reduce the number of feature points visible in an intermediate state, making no detection more likely.	<ul style="list-style-type: none"> - False rotations are still possible, making for a frustrating user experience. - Like type B, rotation is quantized (3 states) unlike type A. This prevents real-time animation of the virtual representation as angular tracking is not possible.
D		The spaces between trackers have been made equal to the size of the trackers. This greatly reduces the number of visible feature points during intermediate states.	<ul style="list-style-type: none"> + This design solves the false rotation detection problem, especially when used with the tracker guard depicted in figure 5.e (marked with a blue X).
E		This design experiments with a larger number of trackers (4 vs 3). The intention was to reduce the distance require when scrolling between selections (states).	<ul style="list-style-type: none"> - Spacing reduction results in false rotations being detected again. - Reducing the width of the trackers to match the spacing reduces their feature points, which also reduces detection reliability.

Selection wheel D in table 5.f. was found to balance the three characteristics above and was chosen as the final tracking surface for the selection wheel of iteration 2.

Iteration 2 of the controller underwent the same expert evaluation as the first iteration of the controller and the following weaknesses were identified:

1. It was difficult to comfortably hold the controller when rotating the selection wheel due to the wheel being exposed at the back of the controller. This is more of an issue than in iteration 1 as the selection wheel is much larger in comparison to the base.
2. It was difficult to determine when the selection wheel was close enough to the camera to enable detection. This is especially confusing as the base's tracker is larger, which enables tracking at a greater distance. This means that it is possible to track the base without being able to track the selection wheel, even though they are physically connected.

The feedback provided by this expert evaluation was used to produce a 3rd iteration of the controller. This version is depicted in figure 5.g and makes the following improvements over iteration 2:

1. The selection wheel stopper was replaced by a larger back-plate in order to prevent the user from accidentally touching the selection wheel. The holes are used for grip.
2. A tracker that is similar in size to the trackers on the selection wheel was added to the selection wheel guard. This tracker is used to determine if the selection wheel is close enough to the camera to detect wheel rotation, since then the base's tracker is detected and the selection guard's tracker is not. A hint can be shown to the user to ask them to bring the controller closer to their face when making a selection.

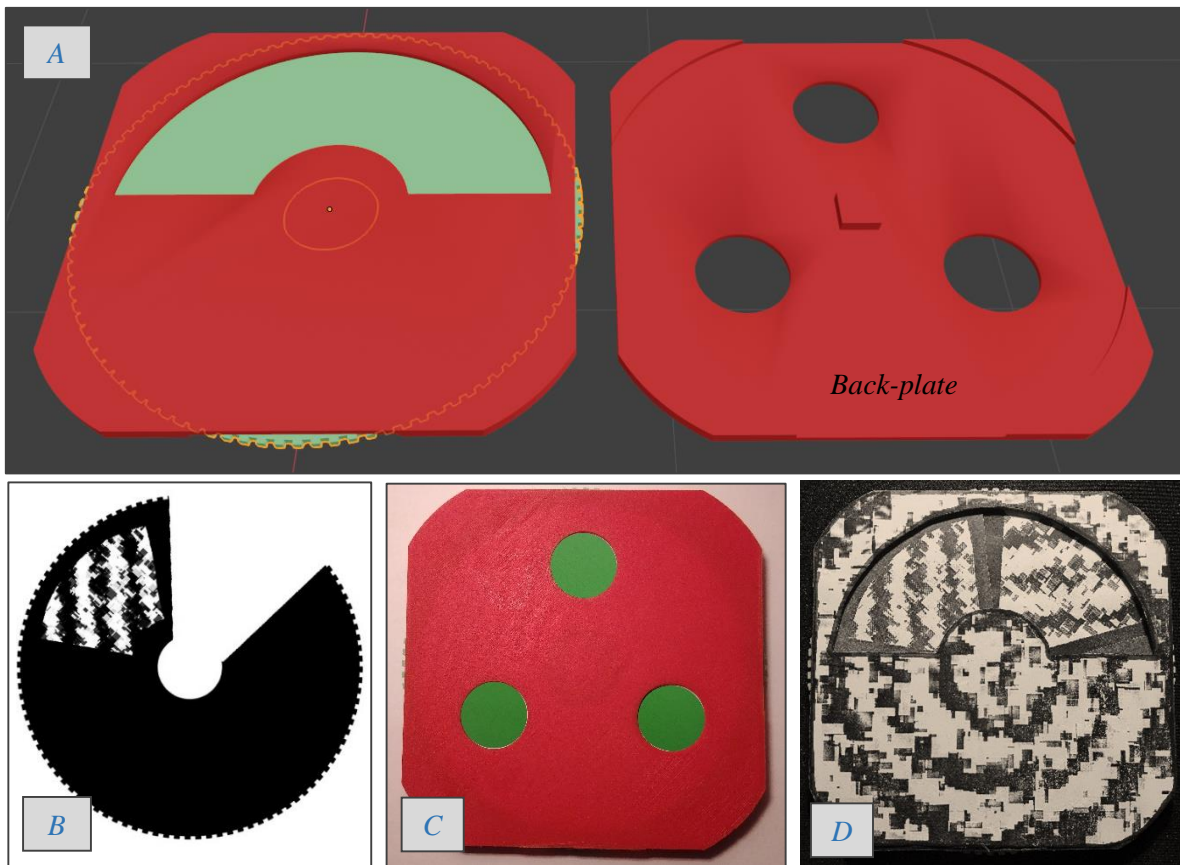


Figure 5.g. Iteration 3 of the High Fidelity Prototype. (A) components in Blender. (B) the new Selection Wheel Guard that replaces the version used in Iteration 2 of the controller. (C) the back view of the controller and (D) the front view with the new selection wheel guard.

Iteration 3 of the controller represents an interface that fulfils the requirements set out in subsection 5.2 when coupled with an appropriate software implementation. For research purposes, two more changes were explored that could potentially enhance the user experience. The two features that were explored are:

1. **Haptic Feedback:** A bump was added to the selection wheel as shown in figure 5.h in an attempt to add haptic feedback to the selection wheel. This is useful because the selection wheel's tracking is quantised, which makes the wheel's position in real the world less clear when in VR. The bump slides into the holes in the back-plate providing haptic feedback. This feature was ultimately a failure as it took too much effort to move the bump out of the hole. This feature was therefore not used in the final version of the controller. Future work could explore a more complex gear system that would allow for more sophisticated haptics.
2. **Back-plate Tracking:** A tracker was added to the back-plate, as shown in figure 5.i, so as to determine when the back face of the controller is in view. Since this tracker is large, this can also be used to accurately track the orientation of the controller. During testing, it was found that this could enhance the user experience by providing another form of input. The addition of this feature to iteration 3 of the controller represents the final version high-fidelity interface.

The final version of the controller takes around 15 hours to 3D print on a Creality Ender 3 and it uses around 82 grams of PLA. This equates to \$1 for electricity and about \$2 for the printing materials. Subsequent sections discuss the software implementation of this interface along with the evaluation environment developed to test this controller.



Figure 5.h. A version of the selection wheel that contains a bump on the back. This bump fits into the holes in the back-plate in order to provide haptic feedback.

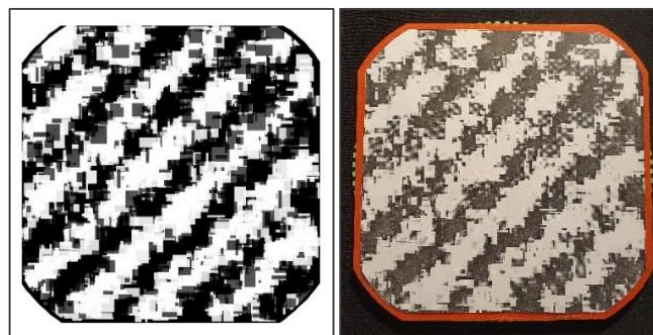


Figure 5.i. The addition of a tracker to the back-plate of the controller. This can be used to detect if the controller is flipped and can provide another form of user input.

5.5. Fauna and Flora Identification Game

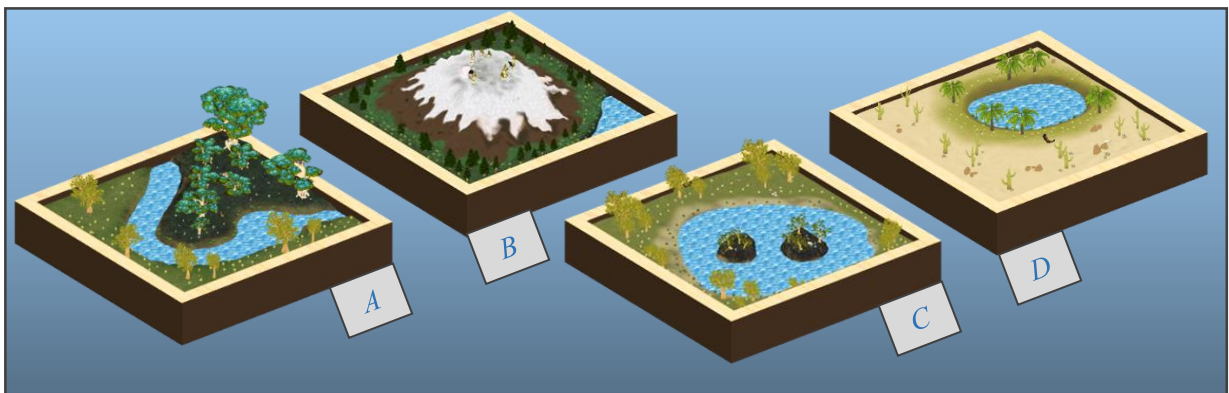
An educational virtual reality experience was developed to evaluate the high-fidelity interface prototype. A fauna and flora identification concept was chosen because of its reliance on visual elements. In this game, users are presented with a miniature virtual terrain and must answer questions about areas within this terrain. These areas are based on the concept of biomes and they include a desert, forest, grassland, tundra and river area. Since this game is only a demonstration of an application of the prototype interface in an educational context, its educational value will not be formally evaluated. Instead, this game is used to evaluate how the prototype interface effects immersion in comparison with a more established interface approach.

The game was developed using the Unity3D game engine due to its cross-platform support. This was useful because while the software is deployed on Android, testing and debugging on a Windows 10 desktop is far more efficient. The following is a complete list of the software technologies that were used to realise the game design:

- **Unity3D v2017.4:** Used to develop and compile the game.
- **Google Cardboard Plugin for Unity:** This plugin handles the head tracking aspects of the virtual reality experience.
- **GearVR Plugin for Unity:** An SDK that allows access to the Samsung GearVR controller. This includes reading the Samsung controller's orientation and detecting button presses.
- **Visual Studio 2019:** Used to write and debug game code.
- **Blender v2.80:** Used to create the terrains and modify purchased 3D assets.
- **Audacity v2.3.3:** Used to edit audio clips.

The following subsections will detail the design of the educational game along with the software implementation of the high fidelity interface prototype.

5.5.1. Game Design



***Figure 5.j.** The four terrains that users may encounter when playing the fauna and flora identification game. The terrains are (A) forest & grassland, (B) tundra & forest, (C) grassland & forest islands and (D) Desert & Grassland Oasis.*

The game takes place in a virtual room as shown in figure 5.k. This room is themed to be similar to the dedicated VR room used for evaluation. The similarities include the lighting style, carpets, perceived room size and walls. The virtual version of the room is enhanced with a view of a surrounding forest and information panels mounted on the ceiling. Lastly, there is a table in the centre of the room that houses the terrains depicted in figure 5.j.

The placement of the terrain in the room allows users to inspect it from above as shown in figure 5.k (A). This terrain table is the main object that users will interact with. There is also a panel mounted to the left part of the ceiling that gives users information about their score and the remaining time for the evaluation. Mounted on the right ceiling is a panel that acts like a map key for the current terrain. It aims to help users understand which part of the terrain a given question relates to. Together, these two panels help to inform users about the current state of the game. Lastly, a grey capsule is placed under the user's view to provide the sensation of having a body [26].

The table in the centre of the room can display one of the four miniature terrains at any given time and each terrain has a unique theming. As shown in figure 5.j, terrain A is a river area, B is a mountainous area, C is a lake area and D is a desert oasis. Additionally, 24 different types of fauna and flora (see Appendix A.1) inhabit these terrains and all fauna are animated to improve realism. These four distinct terrains were developed so that each of the two interface approaches being evaluated (CV versus established controller) could be tested with two different terrains each. The visual differences of these terrains help to make the evaluation process more interesting to users. The way in which this game is used for evaluation is discussed in further detail in the following chapter.

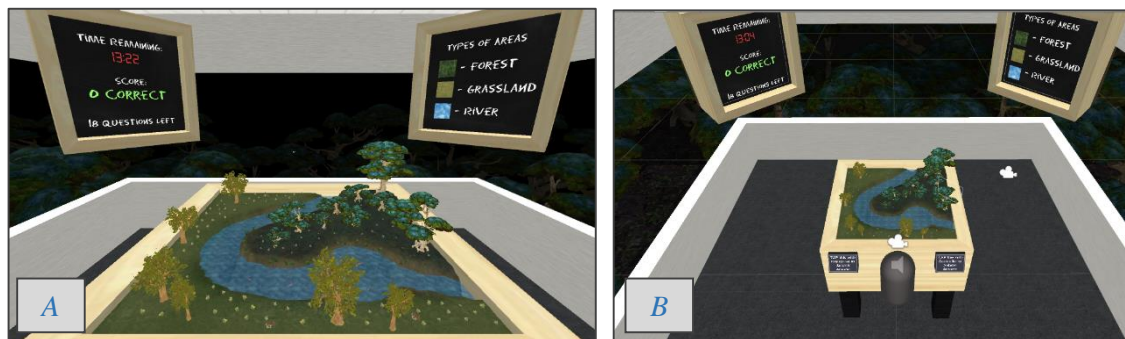


Figure 5.k. (A) the user's view when playing the game and (B) a world view.

In the game, users are asked questions about the current terrain. These questions revolve around what types of fauna and flora inhabit specific areas in the terrain. For simplicity, only 1 type of tree, small plant and animal can inhabit an area and terrains contain 3 areas in total. Users are asked 9 questions about each terrain and they can choose from a selection of three possible answers. They are given 15 minutes to answer the 18 questions of a terrain set and this is visually displayed so that they are aware of the time constraints. Once all the questions have been answered or the timer runs out, users are instructed by the game to remove their HMD.

Another important aspect of this game's design is the software implementation of the controllers. This includes the mechanics of the controller along with visual and auditory cues used to represent it. These details will be discussed further in the next subsection.

5.5.2. Controller Implementation in Software

The high-fidelity controller developed in the previous section represents the physical component of the overall interface. The corresponding software implementation can be viewed as a bridge between the physical prop and the educational game. This software implementation includes the following components:

- **Controller Mechanics:** The Unity Scripts (code) that define the controller's behaviour.
- **Controller Representation:** The 3D models and images that visually depict the interface.
- **Auditory Cues:** The sound clips played when interacting with the controller.

Two software interface implementations have been developed, since two interfaces will be evaluated using this game. The first implementation is for the computer-vision based controller developed in the previous section (depicted in figure 5.1). This has been implemented as follows:

- **Controller Mechanics:** The controller provides the user with the current question along with 3 possible answers. The user must rotate the selector wheel in order to make a selection (e.g., rotate clockwise to select the next option to the right, with modulus rollover). Once the user is satisfied with their selection, they can submit their answer by tapping on the virtual boxes on the front of the table with the controller (Figure 5.1 (C)). The controller can also display hints to show if it is too far away to detect a selection (Figure 5.1 (A)) or if the selector is not found (Figure 5.1 (B)). Lastly, the controller allows for magnification when reversed (Figure 5.1 (B)). In this mode, a magnified view is displayed on the back of the controller and the user can change this view by moving the controller in 3D space.
- **Controller Representation:** The virtual representation of the controller resembles the physical prop for more intuitive mapping. Its colours are understated in order to make selections and text more Visible. Hints are attached to the objects that they relate to in order to make their meaning clearer.
- **Auditory Cues:** Click noises sound when selection changes and answer are submitted.



Figure 5.1. The software implementation of the computer-vision based controller.

In order for the evaluation of the second standard electronic controller to be fair, a second specialised software interface was developed. This controller, formally known as the Samsung GearVR controller (Figure 5.m (A)), has physical buttons and 3 degrees-of-freedom (DOF) tracking. This is different to the computer-vision controller in that there are more physical interaction methods (e.g., buttons), but tracking only supports pointing, not controller localization. The second software interface must therefore leverage these differences in order to create a comparable user experience, which will allow for a fair comparison. This second interface is implemented as follows:

- **Controller Mechanics:** Much like the previous interface, the controller provides the user with a question and 3 possible answers for it. Selection is made by pointing at the desired answer and pressing the trigger on the back of the controller to submit. The pointer is pinned to the side of the user's capsule, since this controller only has 3 DOF tracking. Magnification is also possible and is toggled by pressing the thumb pad on the controller. Since this controller does not support 3D localization, the magnification window is pinned to the user's gaze direction (i.e., head tracking). The surrounding view is dimmed (Figure 5.m, (C)) to make the experience less jarring.
- **Controller Representation:** The interface's virtual representation is made up of two parts. Firstly, there is a model with a pointer visualization to represent the controller's pointing direction. Secondly, there is an information board that is mounted to the centre of the virtual ceiling, which shows question information along with the user's selection. The style of this board is a mix between the style of the other information boards and the computer-vision controller's layout. This was done to improve visibility at a distance, while still standardising the layout to make it more comparable with the other implementation (see Figure 5.m. (B)).
- **Auditory Cues:** Much like the previous interface, sound clips are played when a selection is made and when an answer is submitted.



Figure 5.m. A software implementation of the UI for the Samsung Gear VR controller.

Chapter 6:

Experimental Design

6.1. Introduction

The aim of this project is to design a computer-vision based interface for use in an educational smartphone-driven virtual reality experience. Shute et al. [42] demonstrate that stronger engagement in educational games result in better performance by the participating students. Also, Kato and Miyashita [20] show that virtual reality immersion levels improve when appropriate interfaces are used. This suggests that a virtual reality interface needs to be appropriate for its given application as effectiveness improves immersion, which enhances student performance.

In order to evaluate the effectiveness of the user interface developed, its immersion is measured using the standardised Game Experience Questionnaire (GEQ) proposed by IJsselstein et al. [17]. The GEQ was chosen over the Player Experience of Need Satisfaction (PENS) Questionnaire [40] and the Game Engagement Questionnaire (GEngQ) [6] as it explicitly measures immersion. The GEQ was also chosen over the Immersive Experience Questionnaire (IEQ) [18] as it is a shorter questionnaire (i.e. less items), which reduces participant fatigue by reducing the overall evaluation time needed. Lastly, the GEQ is also a popular choice in VR interface evaluations [21] and the results of these questionnaire options have been shown to correlate strongly [11].

These immersion results are compared to the GEQ results produced by the same users, when using an alternate established electronic controller created by Samsung. This electronic controller has a number of physical buttons and has 3 degrees-of-freedom tracking. Both these interfaces are depicted in figure 6.a. If the immersion results of the computer-vision based interface are equal to or greater than the immersion results of the electronic controller, then it can be concluded that the computer-vision based interface is an effective user interface.

The following chapter will detail the experimental design and the reasoning behind it. The experimental procedure will also be described in detail along with the task design. Copies of the documentation used throughout the user experiment has also been included in Appendix B.



Figure 6.a. Two interfaces being evaluated. On the left is the electronic controller created by Samsung and on the right is the 3D printed computer vision interface developed in this work.

6.2. Experimental Design Overview

The sample size of the user study is 20 university students, as this is approximately the same number of users included in evaluations done in similar work, such as that of Tregillus et al. [49] and Yan et al. [51]. These users were recruited using convenience sampling and users with sensitivity to simulator sickness were screened out to reduce the likelihood of incomplete evaluations. Poster and social media based advertising were used to recruit users and a small monetary incentive was offered to compensate the participants for their time. Lastly, users were warned about the potential of experiencing simulator sickness [28] and they were informed that they could exit the study at any point should they begin to feel any discomfort.

The target population was computer literate university students who are familiar with 3D games. This reduces the time required for orientation and it ensures that users focus on the interfaces being tested. This was ensured using a pre-experiment questionnaire that queried the information listed in table 6.b. and the layout of this questionnaire is given in in Appendix B.2.

Table 6.b. Information queried in the pre-experiment questionnaire.

#	Field	Purpose
1	Age	Used to ensure that users fall within the university student demographic (age 18-28) as the interface being developed is intended for an educational setting.
2	Gender	Used to measure the gender representation in the study. May allow for the identification of unexpected gender biases in the interface's design.
3	Field of Study	Potentially a confounding factor as students in fields of study that work with 3D simulations may be able to learn interfaces more quickly.
4	Experience with 3D games	Potentially a confounding factor as experience with 3D games may improve user performance with familiar controllers.
5	Experience with virtual reality games	Potentially a confounding factor as experience with other virtual reality games may influence overall performance and interface expectations.
6	Experience with educational games	Potentially a confounding factor as previous experience with educational games may influence the user's ability to adapt to the testing environment.

In addition to ensuring computer literacy, the questions regarding past experience in the questionnaire also record potential confounding factors. Past experience may influence how difficult it is for a user to learn a new interface. It is therefore important to record this information in order to allow for confounding factor analysis [4].

This study will make use of a within-subjects design. This is useful for removing the effect of varying virtual experience across users. The drawback of a within-subjects design is the potential for users to learn from their experiences, thus performing better in the second interface trial. This is known as the learning effect [10]. To mitigate this, half the users test the computer vision-based interface first and the second half test the electric controller first.

Users will be evaluating both interfaces in the fauna and flora identification game described in the subsequent subsection. A screenshot of this game is provided in figure 6.c. This evaluation application contains four possible environments with 9 questions each, as shown in figure 6.d.

These four environments are split into two sets and these are alternated between the two interface options. This is done to decouple the difficulty of each environment set and the immersion level of each interface. This means that there are four treatments in this experiment as there are two interface options and two environment sets.

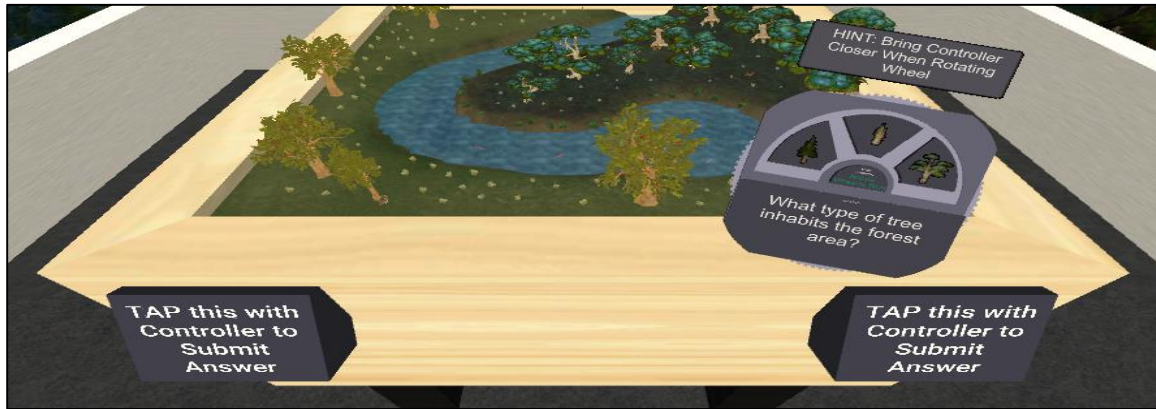


Figure 6.c. A screenshot of the evaluation application, which includes the visualization of the computer-vision based interface and one of the four test environments.

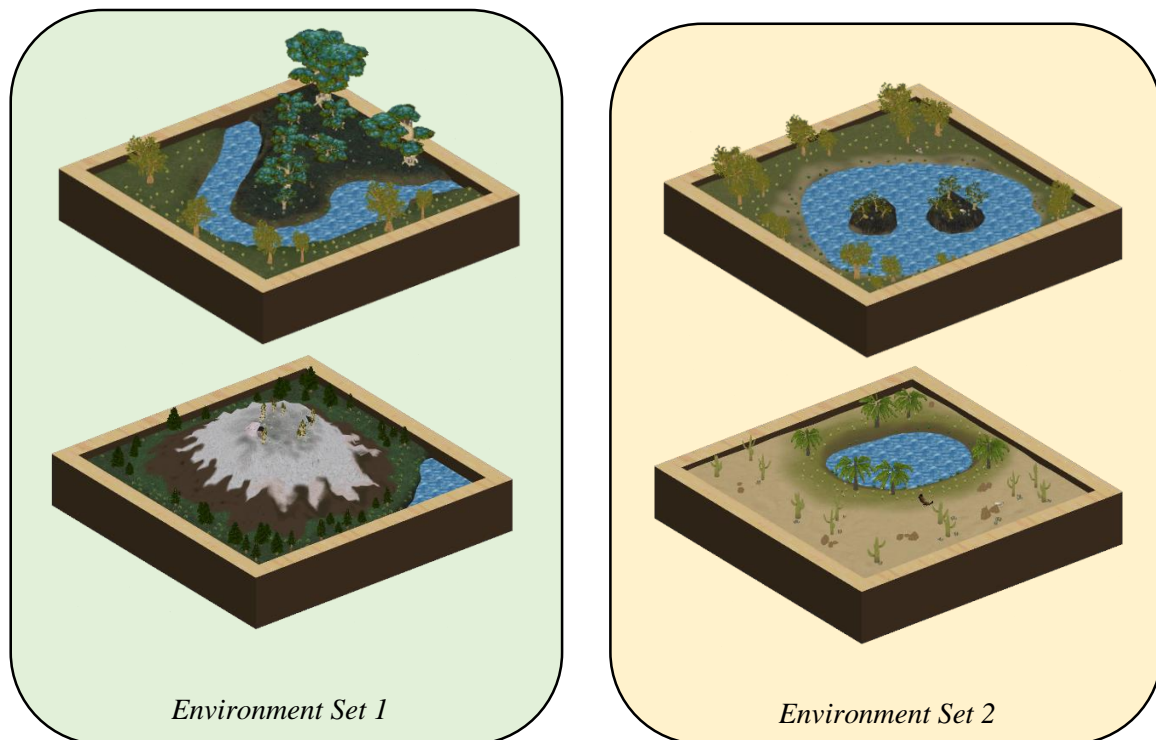


Figure 6.d. Above are the four test environments used in the evaluation application. These are grouped into two sets as shown above to allow for a systematic means of alternating them between interfaces.

The trials were initially conducted in a dedicated virtual reality room to ensure consistent lighting conditions for the computer-vision based interface and access to electricity for charging batteries. This was later changed due to the Covid-19 pandemic as described in section 6.6. The educational application was run on a Samsung Galaxy S8 with a Exynos 8895 CPU and 8GB of Ram. This smartphone also has a 12MP rear camera that is used by the computer-vision based interface. This device was chosen for its compatibility with the Samsung Gear VR headset, a device that is also being used by this project. This VR headset was chosen as it is more durable than the Google Cardboard [44] and this robustness is useful considering that multiple users are being tested.

A quality assurance protocol has been put in place to deal with potential issues that might arise during testing. This protocol reduces the likelihood of producing erroneous or incomplete experimental data, which would need to be discarded. The protocol is outline in table 6.e below.

Table 6.e. Protocol used to reduce the likelihood of producing erroneous experimental data.

#	Problem	Mitigation Strategy
1	The VR headset or the Samsung smartphone becomes greasy due to being touched by multiple users. This poses hygiene concerns and may degrade the VR experience for the participant.	Wipe the equipment with an alcohol-based solution after each use. This includes the smartphone, VR headset and controllers. This will also address Covid-19 related issues.
2	The printed tracking surfaces on the computer-vision based interface become faded after multiple uses. This could reduce the tracking effectiveness, which would result in a poorer user experience over time.	Replace the tracking surfaces after at most 10 uses or if the surfaces show signs of fading. Spare surfaces and the stationary needed to replace them will be kept on hand during evaluations.
3	Batteries in the Samsung controller deplete during an evaluation, potentially effecting the user's performance results and their perception of the interface.	Two sets of rechargeable batteries will be used in the evaluations. These sets will be alternated between experiments with one set being used and the other set being recharged.
4	Shortage of documentation such as feedback questionnaires.	At least three spares will be kept on hand at all times as well as spare stationary to fill them in.

6.3. Task Design

Users are presented with an educational game in which they are tasked with identifying fauna and flora present in a given environment. For each interface type, users are presented with two environments and each environment has 9 questions associated with it. Each question is a multiple choice question and contains 3 choices, each with a name and an illustration. Users are given 15 minutes to answer all the questions. An example of how these questions are presented is provided in figure 6.c and this example relates to the forest area. Examples of these questions are as follows:

- What type of tree inhabits the forest area?
- What type of small plant inhabits the grassland area?
- What type of fish inhabits the lake?
- What type of animal lives in the forest area?
- What type of bird soars in the skies?

These questions have been designed to vary in difficulty. For example, questions about trees are the easiest as there are many trees in each environment and trees are the largest type of object being identified. In contrast, questions about fish and birds are more difficult as there are fewer of them in the environment, they are relatively small and they move around. Users may need to make use of the magnification abilities of the interfaces in order to be able to answer these more difficult questions.

6.4. Measures

Both qualitative and quantitative information will be used to evaluate each interface. Qualitative information will be collected from users in the form of questionnaires. This is specifically the standardised Game Experience Questionnaire (GEQ) proposed by IJsselsteijn et al. [17]. Given the nature of this experiment, only the Core Module was used from the questionnaire and question 3 of the module was also omitted. The Social Presence module was excluded as there are no social elements in the evaluation. The remaining modules were left out as users only complete the questionnaire after each experience. This results in a questionnaire with 32 items. In addition to these questions, users are also presented with the opportunity to give positive and/or negative feedback regarding their overall experience. This will allow users to specify aspects of the experience that stood out to them.

To enable further analysis of the qualitative information collected, quantitative data will also be automatically recorded while users are conducting the evaluation. This quantitative data is a variety of timestamped performance metrics including:

- If a user answered a question correctly or not.
- How long it took for users to answer each question.
- When users used the magnification feature.
- Various interactions specific to each type of interface (e.g., when a selection was made, when a selection was changed, ...).

Together these two forms of measurements will allow for the contrast and comparison of the two interfaces being tested. The electronic controller created by Samsung is expected to perform well as it is an established and conventional interface approach. If the computer-vision based interface approach performs similarly or better than the electronic controller, then it can be concluded that it is an effective interface approach as it performs comparably to a widely accepted approach. If the computer-vision controller performs worse than the electronic controller, then the qualitative and quantitative information will be used to further understand why the interface approach performed poorly.

6.5. Experimental Procedure

The experimental design introduced in the previous section is realised in a series of six steps. Each step represents a distinct activity for the user being evaluated and these steps are:

1. The participant is shown a set of introductory slides (Appendix B.4) containing information about the broad research topic, the evaluation software's objectives and information about simulation sickness. They will then be asked to sign a consent form (Appendix B.1) to enable participation in the study. This activity requires at most 10 minutes.
2. The participant is provided with a set of tutorial slides (Appendix B.4) to show them how to use the first interface method. They then use the selected interface method to answer 18 questions across two terrains in a game (Appendix B.3). Performance metrics will be collected while the participant is playing the game. This includes information about participant actions and the amount of time taken to complete each question. This activity runs for at most 15 minutes.
3. The participant will then be asked to complete the Game Experience Questionnaire (GEQ). The version of the GEQ used in this study only makes use of the core module and discards the question relating to story. This activity is run for at most 10 minutes.
4. The participant is shown another set of tutorial slides to show them how to use the second interface method. They then use the interface method to answer another set of 18 questions across two different terrains. The same metrics as in step two are collected. This activity is run for at most 15 minutes.
5. The participant will then be asked to complete the GEQ for the second interface method. This activity requires at most 10 minutes.
6. Upon successful completion of the previous steps, the participant is compensated for their time. They will be required to sign a document to acknowledge receipt of the compensation. This activity is run for a couple of minutes.

The overall evaluation procedure takes roughly 1 hour. This length of time is long enough to allow users to adequately test each interface, while also being short enough to reduce the likelihood of fatigue.

6.6. Extension to Enable Testing under Covid-19 Pandemic

During the course of the user evaluation, the global Covid-19 pandemic started. In response to this, the country implemented a lockdown procedure to restrict in-person contact. Given that the evaluation was designed for execution in a dedicated University VR room, it needed to be updated to allow for the safe and remote execution (i.e. testing at home) by participants. This subsection details all the changes that were implemented to make this possible.

The first step was to develop a means of packaging the evaluation equipment so that it is portable, easy to disinfect and easy to understand. A custom package was created (see figure 6.f) using plastic that is durable and easy to sanitise. The plastic chosen was also transparent so that users could easily find the components needed for the evaluation process. An alcohol based sanitiser and a cleaning cloth was also included in the package so that users were empowered to sanitise as needed.



Figure 6.f. A package containing the evaluation equipment. It contains the electronic controller, the vision controller, the VR headset (with smartphone pre-docked) and sanitisation items. The delivered package also contained a pen, instruction sheets and questionnaire forms.

Some software upgrades were also needed so that the project could facilitate remote evaluation. First, the smartphone's home-screen was simplified (see Figure 6.g (A)) so that users could conduct the evaluation on their own using a procedure guide (Appendix B.5). The evaluation software was also upgraded so that users could select an anonymous profile (see Figure 6.g (B)) based on a pre-supplied identification number (i.e. 1 to N). Selection was done using a timed gaze approach, so that participants would not be partial to a specific controller before the experiment started. Only expected user profiles were visible at any given time to simply the experience. This was done so that multiple users could be tested remotely at once (e.g., if a couple was being evaluated). The last change was the addition of an in-game voice prompt that sounded after a user selects their profile. This prompt orientates users by reiterating information provided by the introduction content.

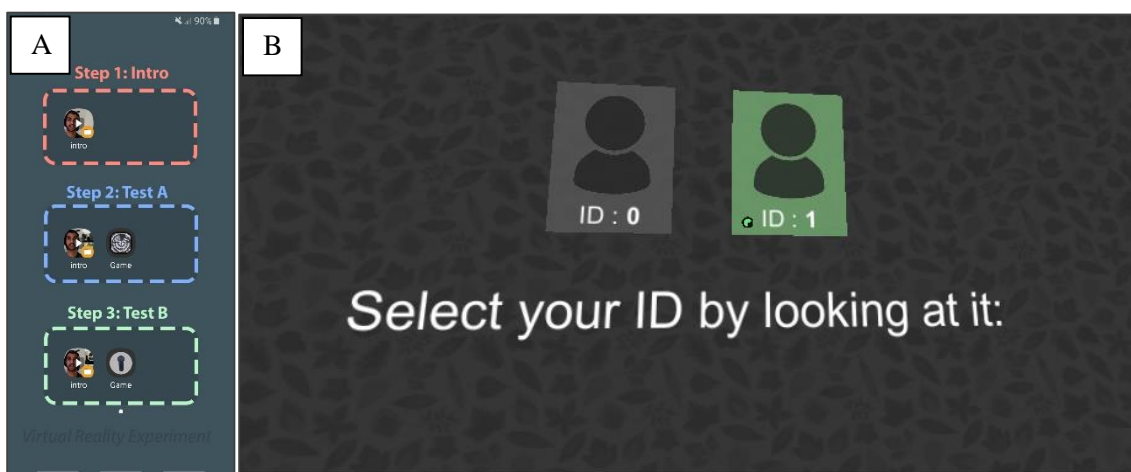


Figure 6.g. (A) the smartphone home-screen which was designed so that participants could more easily find testing content. (B) the profile selection screen added to the game.

The last step in preparing the evaluation package was to create video versions of the introduction slides (Appendix B.4) as well as instructional documents (Appendix B.5, B.6). This was necessary as a researcher would not necessarily be present during an evaluation due to social distancing restrictions. It should be noted that participants were offered the option of having a video call so that questions could be answered immediately.

In addition to these upgrades, a sanitation protocol was also implemented to protect against Covid-19 infections. This protocol is as follows:

1) Pre-Experiment Process:

- a) User Screening: the user is asked if they have experienced any flu-like symptoms and if they had contracted Covid-19 or been in contact with anyone who has.
- b) The user completes the consent and demographics forms online using JotForm⁸ to reduce physical interaction.
- c) The evaluation package and included items are sanitised using an alcohol based sanitizer and new documents are printed for the user.

2) During-Experiment Process:

- a) The package is delivered to the participant's home. Both the participant and the researcher wear facial masks during the exchange and social distancing is respected.
- b) The researcher may stay with the participant if both parties are comfortable with this and in-person testing is carried out using masks and social distancing practises.
- c) The researcher may also use a video calling tool of the user's preference to allow for true remote testing.

3) Post-Experiment Process:

- a) The package is retrieved from the participant's home using a face mask and the package container is sanitised.
- b) The package is then sanitised in its entirety and all documentation is stored in a container for processing at a later date.
- c) The sanitation process damages the vision controller's tracking surfaces, which need to be replaced after 3-5 rounds of testing. The phone and electronic controller's batteries are also recharged after each evaluation.

These changes do not result in any significant changes to the experimental procedure described in section 6.5. The main difference is that users watch instructional videos instead of listening to in-person slide presentations. This has the benefit of making the testing procedure more consistent, as all users receive the exact same information before evaluating a controller.

⁸ <https://www.jotform.com/>

Chapter 7:

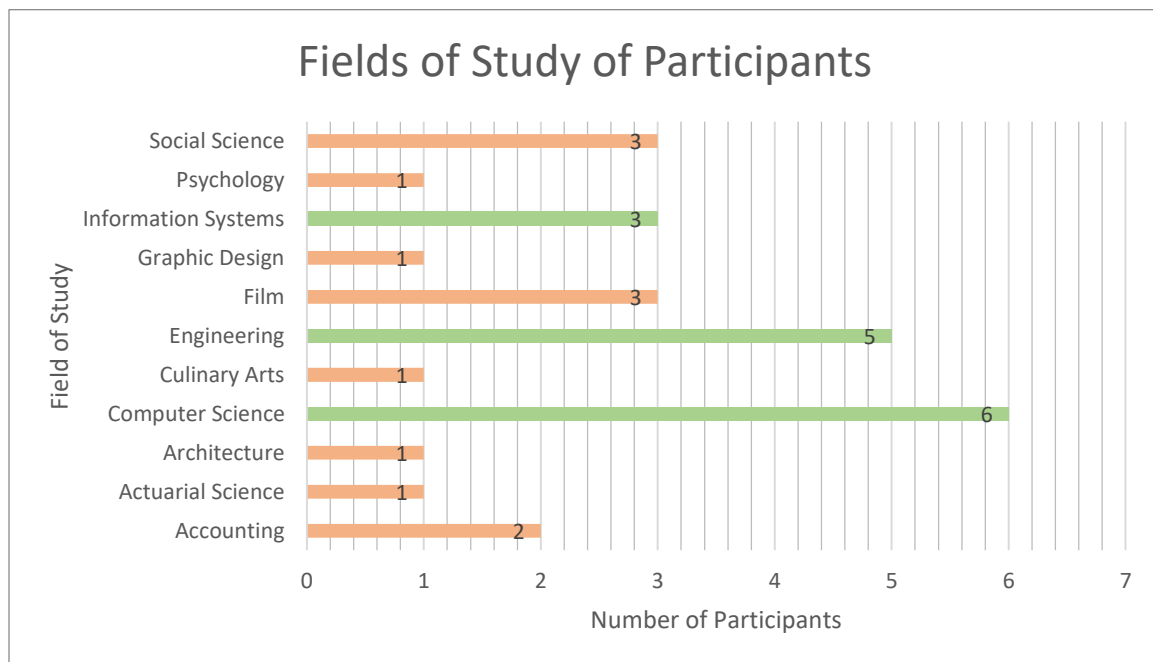
Results & Discussion

This project aims to investigate whether a computer-vision based interface can be as immersive as a conventional electronic controller. The previous chapters detail the development of this interface and the educational game used to evaluate it. This chapter examines the results of this evaluation by analysing both the quantitative and qualitative data generated by it.

7.1. Demographics

The user evaluation of the two controllers was carried out with 27 participants. This participant count was considered to be sufficient as it is comparable to other similar studies on virtual reality interfaces. Specifically, Gugenheimer et al. [15] (18 participants), Tregillus et al. [49] (25 participants) and Tregillus and Folmer [48] (18 participants).

Of these 27 participants, 10 identified as female, 16 identified as male and 1 identified as other. There were 17 users that were right handed and 10 users that were left handed. The youngest participant was 20 years old and the oldest was 34 years old. The average age was 24 years, meaning that participants could be described as senior university students. Participants were from a range of different fields as shown in Graph 7.a. This is useful as the learning application is not specific to any field of study and this study is more focused on learning mechanics rather than educational content. There is a balance between users that make extensive use of computers (14 users) in their daily activities and those that do not (13 users).

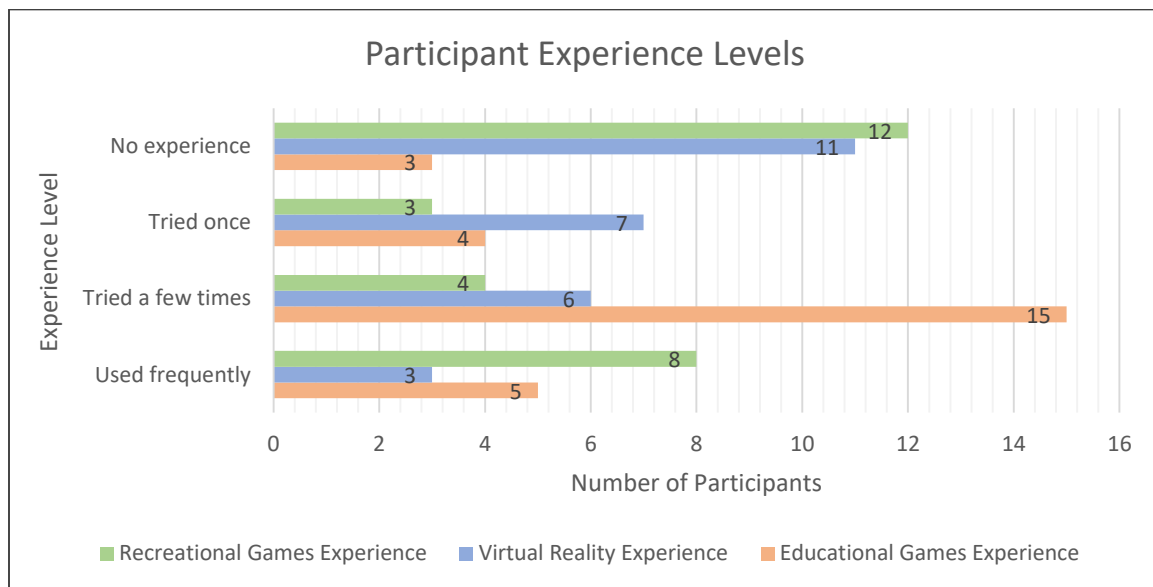


Graph 7.a. A graph showing the representation of users from different fields of study. Technical fields that make extensive use of computers are shown in green ♦, while non-technical fields are shown in orange ♦.

There was also a range of participant experience with recreational games, virtual reality (VR) and educational games as shown in Graph 7.b. In terms of experience with recreational games, most users were not frequent gamers, though there was an almost even split between those with less (no experience and tried once) and more (tried a few times and used frequently) experience. This means that the study was exposed to both users that were and were not familiar with recreational games.

Around a third of participants had no experience with virtual reality and a further 7 had tried it only once. This means that the majority of participants had at most one experience with an immersive environment. This is considered to be a representative spread of experience as VR is not yet a main stream learning technique, meaning that many users of a potential VR learning solution will experience VR for the first time with that solution. It is therefore useful that the results of this study to reflect a majority of inexperienced VR users.

In terms of educational game experience, many participants (20 out of 27) had multiple experiences with educational games (see Graph 7.b). This means that participants in general had familiarity with learning mechanics such as multiple choice questions and immediate feedback. General university students will likely match this majority class as they are familiar with online class quizzes, which share these learning mechanics.



Graph 7.b. User experience across Recreational Games, Virtual Reality and Educational Games as indicated by them on the screening questionnaire.

After completing the evaluation, only one user reported a simulator sickness symptom (ID 20), specifically, feeling “a little unsettled”. Having only one minor simulator sickness incident is a favourable result and it was likely due to the framerate of ~50 FPS for the electronic controller version of the game and ~30 FPS for the computer vision version. Lower framerates would be more likely to induce simulator sickness because the user’s view would update too slowly and not match the motion sensed by them [28, 29].

Lastly in terms of accessibility, the evaluation was exposed to 1 participant with poor peripheral vision in the left eye (ID 4) and 3 users who wear glasses daily (ID 4, 16, 20).

7.2. Game Experience Questionnaire Results

Participants completed a Game Experience Questionnaire (GEQ) [17] after testing each of the two interfaces (see Figure 7.e.). The GEQ offers a number of metrics to compare interface experiences across users. These metrics are Competence, Immersion, Flow, Tension, Challenge, Negative Affect and Positive Affect. These metrics are calculated as an average of their sub-questions (e.g., I felt content) each of which is a rating on a scale between 0 (not at all) and 4 (extremely) [17].

The first step in analysing these metrics is to determine if they are normally distributed as this will determine whether parametric or non-parametric methods are appropriate. The Shapiro Wilk test was used to test each metric for normality and the null hypothesis for this test is that the sample is normally distributed. Additionally, a significance value of 5% was used meaning that if the p-value for a given test is lower than 0.05, then the null hypothesis is rejected and the data is not normally distributed. This significance value is used for all statistical tests that follow. The results of these tests are tabulated in Table 7.c.

Table 7.c. A table of results for the Shapiro Wilk Test for Normality for the GEQ metrics. EL means electronic controller and VC means vision controller.

Metric	Interface	Shapiro-Wilk Normality Test		Is Normal?
		P	W	
Competence	EL	0.003	0.868	No
	VC	0.058	0.927	Yes
Immersion	EL	0.015	0.903	No
	VC	0.053	0.925	Yes
Flow	EL	0.134	0.942	Yes
	VC	0.033	0.917	No
Tension	EL	0.000	0.484	No
	VC	0.000	0.705	No
Challenge	EL	0.077	0.932	Yes
	VC	0.004	0.874	No
Negative Affect	EL	0.000	0.491	No
	VC	0.000	0.729	No
Positive Affect	EL	0.000	0.789	No
	VC	0.011	0.896	No

As can be seen in table 7.c, only 4 of the 14 metrics are normally distributed and none of these 4 are in the same metric set. Therefore, given that it cannot be assumed that the data is normally distributed, non-parametric testing is appropriate for this scenario.

The design of the study must be examined to determine which non-parametric test is most appropriate. This study has one independent variable with two categories (the electronic controller and the vision controller). It also makes use of a within-subjects design, meaning that measurements for each metric type of the GEQ occurs in paired samples for each user. The Wilcoxon Signed Rank (WSR) Test is therefore the most appropriate non-parametric test for this type of study [50]. The WSR Test's null hypothesis is that two given samples are from the same distribution. A 5% significance value was chosen as it is common in this domain and this means that if a p-value is less than 0.05, then the null hypothesis is rejected and the data is statistically different.

Table 7.d. A table of results comparing GEQ metrics across controller types using Wilcoxon Signed Rank Test. EL means electronic controller and VC means vision controller.

Metric	Interface	Wilcoxon Signed Rank Test		Is Statistically Different?
		P	V	
Competence	EL VC	0.144	201.5	No
Immersion	EL VC	0.120	70.5	No
Flow	EL VC	0.625	121.5	No
Tension	EL VC	0.106	18	No
Challenge	EL VC	0.002	30	Yes
Negative Affect	EL VC	0.142	29	No
Positive Affect	EL VC	0.148	107.5	No

After carrying out the Wilcoxon Signed Rank Test on the GEQ metrics across controller types as shown in table 7.d, it was found that the controllers only differ in terms of Challenge. To determine how Challenge differed across controllers, the WSR test was reconfigured to have a null hypothesis that assumes that the median of the distribution that the electronic controller's values are drawn from is left shifted in comparison to that of the vision controller. In other words, this would mean that the median value of the electronic controller's challenge metric is statistically less than that of the vision controller. After carrying out this test, a p-value of 0.001 ($V=30$) was found confirming that only the Challenge metric of the vision controller was statistically greater than that of the electronic controller. In other words, participants found that the vision controller was more challenging to use. Figure 7.e. and Table 7.f. further specify this difference in terms of the metric scores and it can be seen that the electronic controller has almost no challenge for users, while the vision controller has a low challenge rating.

It is also useful to consider the Negative Affect metric in conjunction with the Challenge metric as this would suggest whether the challenge added by the controller was productive or not. Specifically, a task needs to balance skill and challenge to be engaging (i.e., induces flow [19]), meaning that challenge does not necessarily decrease engagement. If the challenge significantly out-weighs the ability of a user can it cause negative emotions like frustration. According to Table 7.f, the vision controller had no negative affect in its interquartile range. This suggests that the challenge introduced by the vision controller did not necessarily negatively impact the user experience. This hypothesis will be further investigated by analysing performance metrics and user feedback in the following subsections.

The final metric of interest is the Immersion metric as this is the main topic being investigate by this project. In general, both interfaces achieved the same "high" immersion rating as shown in Table 7.d. and Table 7.f. This is valuable in two main ways. Firstly, the vision controller provided a comparable amount of immersion in comparison to the electronic controller in terms of GEQ performance. Secondly, both controllers achieved a high immersion rating meaning that they both contributed positively to the overall immersion of the educational application. The following subsections will further investigate the nature of this immersiveness by considering user feedback.



Figure 7.e. Box & Whisker Plots of the GEQ Metrics

Table 7.f. A table GEQ Metrics classed by Score value. This can be seen as an interpretation of the GEQ score. A value between 0 & 1 is a None Class, a value between 1 & 2 is a Low Class, a value between 2 & 3 is a Medium Class and a value between 3 & 4 is a High Class.

Metric	Interface	Lower Quartile Class	Median Class	Upper Quartile Class
Competence	EL	Medium	High	High
	VC	Medium	High	High
Immersion	EL	Medium	High	High
	VC	High	High	High
Flow	EL	Medium	High	High
	VC	Medium	High	High
Tension	EL	None	None	None
	VC	None	None	None
Challenge	EL	None	None	None
	VC	None	None	Low
Negative Affect	EL	None	None	None
	VC	None	None	None
Positive Affect	EL	High	High	High
	VC	High	High	High

7.3. Performance Metrics

7.3.1. Battery Consumption

One way to compare the vision controller and electronic controller is by considering their power consumption as this may be important for real world applications. The more power an approach consumes, the more frequently the device will need to be recharged. This would affect the way the application is used in a classroom, as instructors would need to factor in charging time and the related charging cost. The battery consumption was noted at the beginning and end of each evaluation for all 27 participants and using the Shapiro Wilks test for normality, it was seen that both the vision controller ($p=0.018$, $w=0.905$) and the electronic controller ($p=0.003$, $w=0.867$) had battery consumption that was not normally distributed. This data is plotted as a box and whisker plot in Figure 7.g. and given how these two distributions compare visually, a WSR test was used to compare them.

The WSR test was configured to have a null hypothesis that assumes that the vision controller's battery consumption is greater (i.e., right shifted) than that of the electronic controller. The WSR test resulted in a p-value of 1 and a v-value of 0. This suggests that the median consumption (6%) of the vision controller is statistically more than the consumption of the electronic controller (3%). Given that the Samsung Galaxy S8 used in this study has a 3000mAh battery, this consumption translates to an average consumption of 180mAh and 90mAh, respectively.

This result suggests that the computational cost and/or usage characteristics of the vision controller has resulted in double the battery consumption, meaning that the electronic controller can support double the amount of usage before recharging is required.

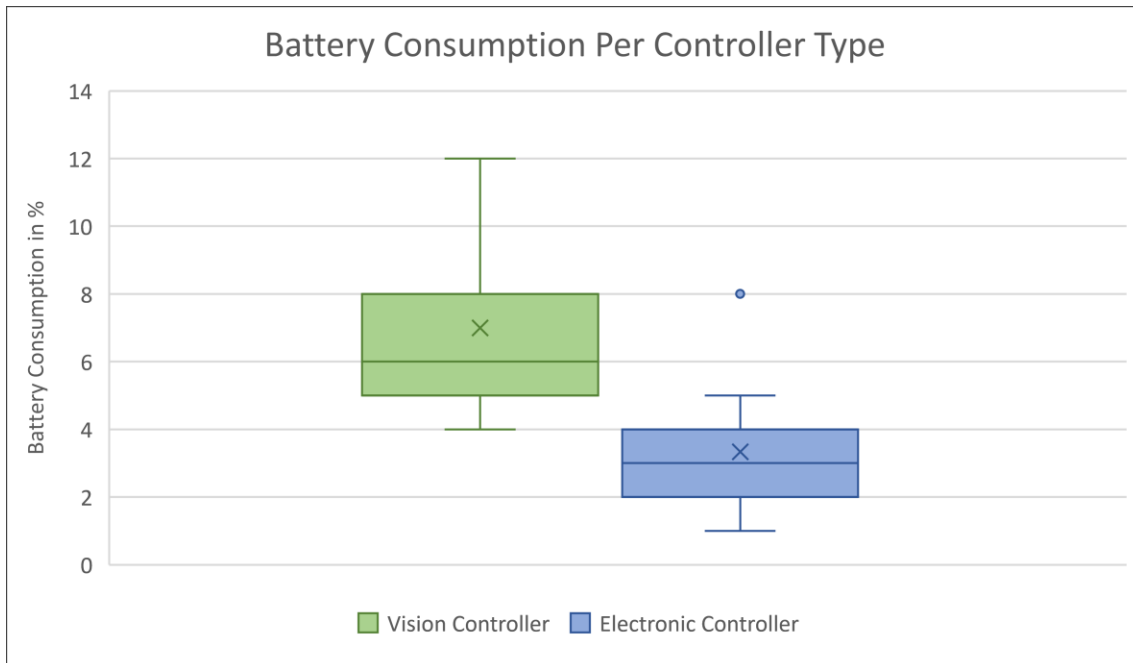


Figure 7.g. A box a whisker diagram showing batter consumption across controller types.

7.3.2. Time Performance

Another metric to consider when comparing controller types is the amount of time taken by participants to complete an evaluation with a given controller. It is valuable to examine this metric as it can provide insight into controller efficiency and ease of use. This metric is also useful in designing educational content that needs to be time boxed for use in lessons or workshops. Differences in usage times could also help to explain why the vision controller consumes more power.

The overall evaluation times were recorded for each of the 27 participants across both controllers and this data is visualised in Figure 7.h. Using a Shapiro Wilk Test, it was found that the vision controller's evaluation times were normally distributed ($p=0.314$, $w=0.960$), while the electronic controller's evaluation times were not normally distributed ($p=0.005$, $w=0.879$). A non-parametric WSR Test is thus more appropriate, given that the electronic controller's evaluation times are not normally distributed. The WSR Test was configured to have a null hypothesis that assumes that the vision controller's evaluation times are greater than that of the electronic controller (i.e. right-shifted).

The WSR Test ($p=1$, $v=0$) found that the median (9 minutes) of the vision controller's evaluation times were statistically longer than that of the electronic controller (5 minutes). This appears to correlate with GEQ findings in which participants reported that the vision controller was more challenging, as a more challenging controller would be less efficient. Additionally, it appears that the usage characteristics of the controllers are important in understanding the power consumption of the two approaches (section 7.3.1) as users took almost twice as long in vision controller evaluations, which would cost double the power assuming processing costs we roughly equivalent.

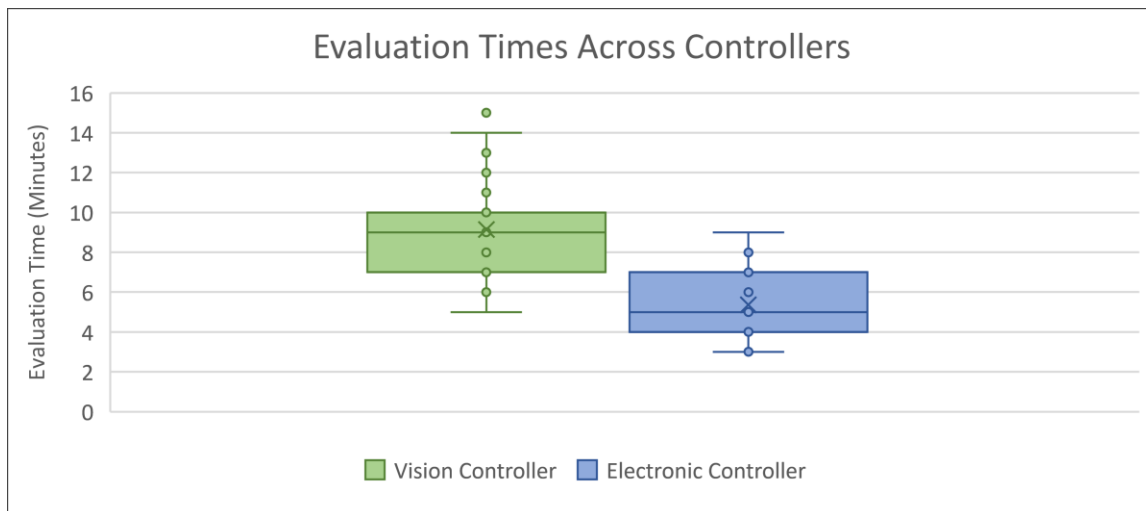


Figure 7.h. A graph showing how long participants took to complete each of the evaluations.

It is also useful to examine the per question time performance of users, in order to further understand if this difference in evaluation times is related to a learning cost or inherent to controller design. If the time difference is mainly related to the controller's design, performance should remain consistent regardless of testing order. For the remainder of this subsection, the overall user experience will be examined as either a vertical split of the experiment (i.e., first 18 questions seen vs. last 18 questions seen, regardless of controller used) or a horizontal split (i.e., vision controller vs. electronic controller, regardless of testing order).

It is worth noting that differences in evaluation times could be explained by learning-related time costs. There could be a time cost related to learning the game, which includes learning to use VR and learning the game rules. There could also be a time cost related to learning to use each interface approach. For this experiment, it is important that the game related time cost be the same regardless of which controller is used first. If this is true, it suggests that the user on-boarding succeeded in preparing users for the game, by minimising the time penalty of learning the system (i.e. learning effect). If the on-boarding was unsuccessful or the game was difficult to learn, we would expect to see that the first interface experienced would always take longer to use. To investigate this, the question times were organised into a vertical split such that the first 18 questions experienced by a user were compared to the second 18 questions experienced by the same user, regardless of which controller was used. This data is shown in Figure 7.i. and it appears that the two trend lines mostly overlap.

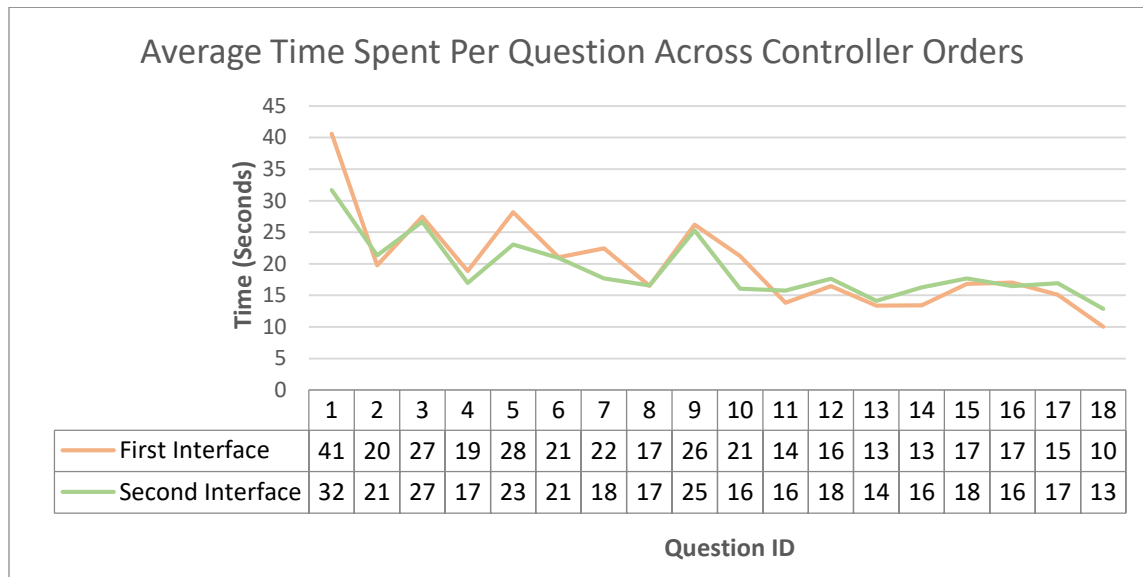


Figure 7.i. *The time users spent per question on the first half of the experiment (18 questions) versus the time users spent per question on the second half of the experiment.*

A Shapiro Wilk's Test for normality was used to further understand the data presented in figure 7.i. It was found that both the first interface's question times ($p=0.042$, $w=0.892$) and second interface's question times ($p=0.017$, $w=0.868$) were not normally distributed. A non-parametric WSR Test was then used to compare these distributions and it was found that the medians of the underlying distributions are the same ($p=0.733$, $v=94$). This suggests that there is no statistically significant difference between these distributions (median=17 seconds) and this means that users did not perform more quickly in the second interface experience. This suggests a small learning curve for the overall evaluation environment.

The 9 second difference between the first question of the first interface experienced and the first question of the second interface experienced is the biggest difference in question times. This can be seen as a time cost related to becoming familiar with the game. This relatively large difference appears to be unique to the first question as subsequent question time pairings have a maximum difference of 5 seconds. It is speculated that the video introductions and initial in-game voice prompt helped to reduce the in-game time cost of learning to use the game and experiencing one submission example cements this knowledge.

Now that it has been confirmed that interface ordering was not a confounding factor, it is possible to examine the time cost per question across the two controllers. This is a valuable metric to examine as it provides insight into how the controllers are used. The times spent per question across controllers is illustrated in Figure 7.j and the trend lines suggest that the vision controller takes more time to use in general as there is a constant vertical difference between trend lines.

The normality of the data depicted in Figure 7.j. was checked using the Shapiro Wilk's Test for normality in order to examine this trend line hypothesis further. It was found that that the per question time of the vision controller is not normally distributed ($p=0.007$, $w=0.845$), while the electronic controller per question time is normally distributed ($p=0.104$, $w=0.915$). Given this result, a WSR Test was used to compare the two distributions and the null hypothesis was configured to assume that the vision controller's distribution is right shifted. The test returned a p-value of 1 ($v=0$) meaning that the per question time for the vision controller is statistically longer than that of the electronic controller.

For the vision controller, participants had a median value of 23 seconds across all questions, while the electronic controller had a median value of 13 seconds. This suggests that the electronic controller's design is more efficient to use as participants took 77% longer to answer the same questions with the vision controller. This result is a quantification of the time cost inherent to the vision controller's design that helps to further explain why participants felt that the vision controller was more challenging to use. More formally, the vision controller required more time to perform the same functions.

This difference in time usage could be explained by the difference in selection (gear rotation vs click), magnification (flip vs click) or answer submission (move and tap vs click) mechanics and this will be examined further in the following subsections.

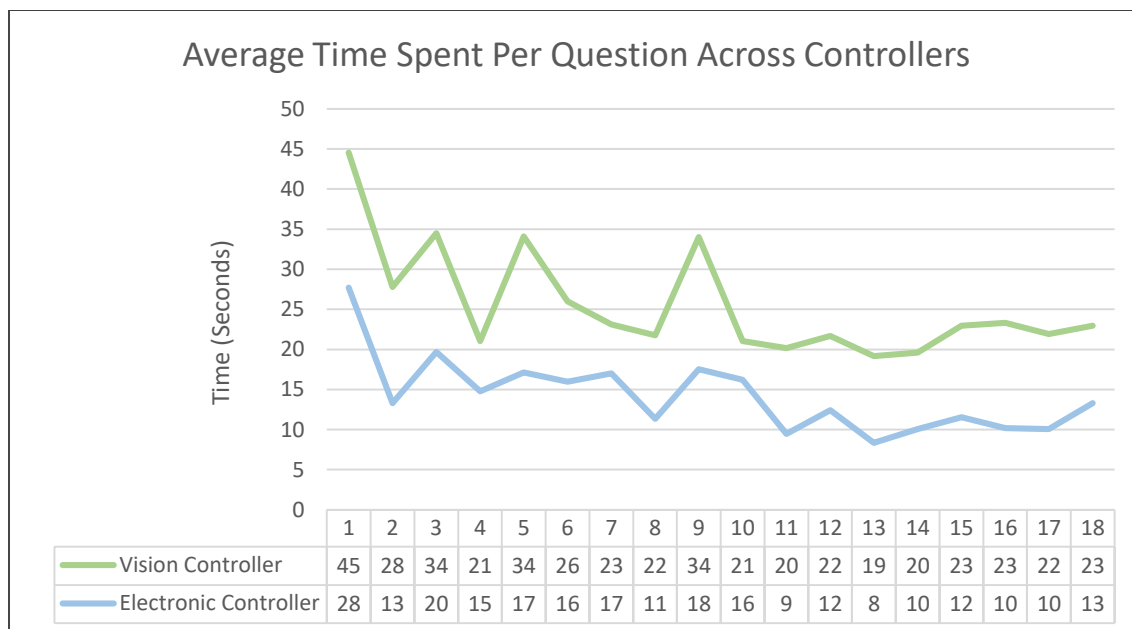


Figure 7.j. The average time spent per question across controller types. The trend lines suggest that the vision controller takes longer to use in general.

7.3.3. Score Performance

The previous section demonstrated quantitatively that the electronic controller is more efficient to use in comparison to the vision controller. While this finding is valuable for characterising controller usage, it is also important to examine user performance in terms of question scores. Specifically, this section aims to examine if there are any differences in the controller approaches in terms of their effect on answer accuracy. Much like the previous section, this section will examine score performance across controller orders and controller types.

During the course of the experiment, users were presented with 18 multiple choice questions for each of the two controller approaches being evaluated. Each question could either be correct (score of 1) or incorrect (score of 0). The average for each question is calculated by adding the score across users and averaging by the total number of users.

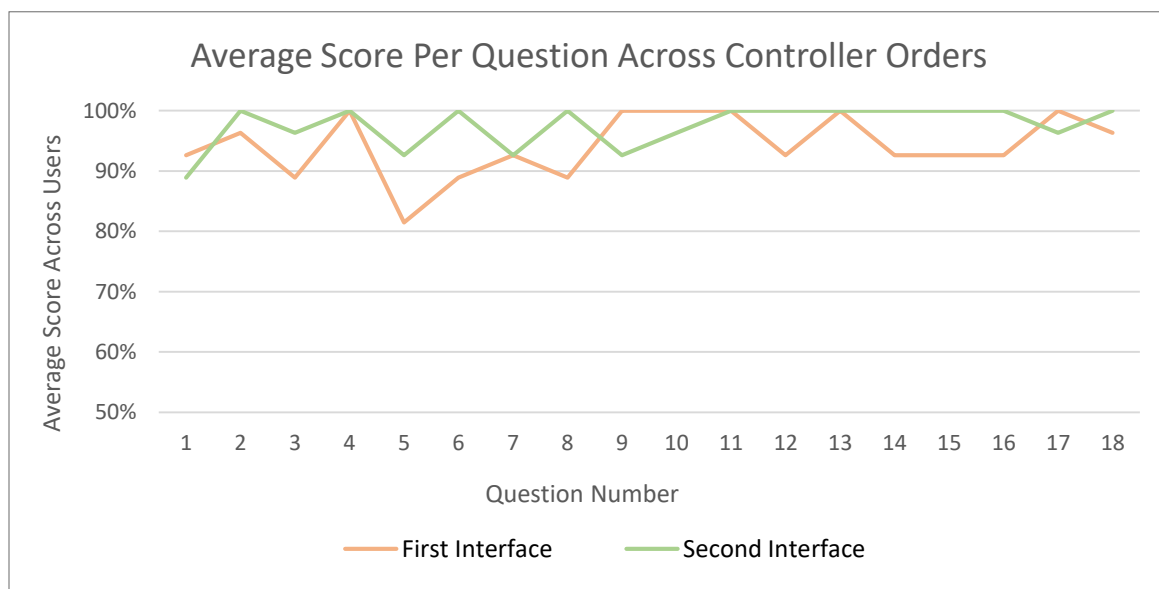


Figure 7.k. A graph comparing the scores of the first interface experienced versus the scores of the second interface experienced.

The average score per question order should be evaluated first to see if ordering had an effect on question score performance (see Figure 7.k). A Shapiro Wilks Test of normality was conducted and it was found that the question score performance of both the first ($p=0.0196$, $w=0.873$) and second ($p=0.0001$, $w=0.720$) interface approaches were not normally distributed. A WSR Test was then conducted to compare these distributions and it was found that they are statistically different ($p=0.028$, $v=17.5$). This means the median value of 93% of the first interface experienced is statistically smaller than the median value of 100% of the second interface used. This result could be explained by the qualitative feedback given by users in the following section 7.4. Specifically, many users did not understand the concept that the map was split into areas and that questions were related to these areas. As a result, many participants answered the first few questions wrongly, which meant that the first interface used would have a lower score than the second as users better understood the concept of areas by the second play through.

The score across controller types (see Figure 7.1.) is a more interesting metric to analyse as it will show if the difference in controller efficiency has score implications. While it was previously shown that users achieved better results in the second interface used, the effect of ordering should be negligible in the per controller analysis, given that 13 users used the vision controller first and 14 users used the electronic controller first due to the permuted order design of the experiment. The normality of question score per controller type was assessed using a Shapiro Wilks Test and it was found that the vision controller ($p=0.002$, $w=0.802$) and electronic controller ($p=0.009$, $w=0.853$) per question scores were not normally distributed. A WSR Test was then used to confirm the null hypothesis that these distributions are not significantly different ($p=0.83$, $v=42$) as is suggested visually by Figure 7.1. This result is consistent with the overall performance of the users as there were 467 and 465 correct submissions by vision and electronic controller users, respectively. This suggests that the efficiency differences of the controllers did not have an effect on the users' performance, both in terms of information clarity and accidental submissions. This result may have been different if the evaluation's time limit was less than 15 minutes as a lower controller efficiency could result in participants not being able to answer all the questions in time.

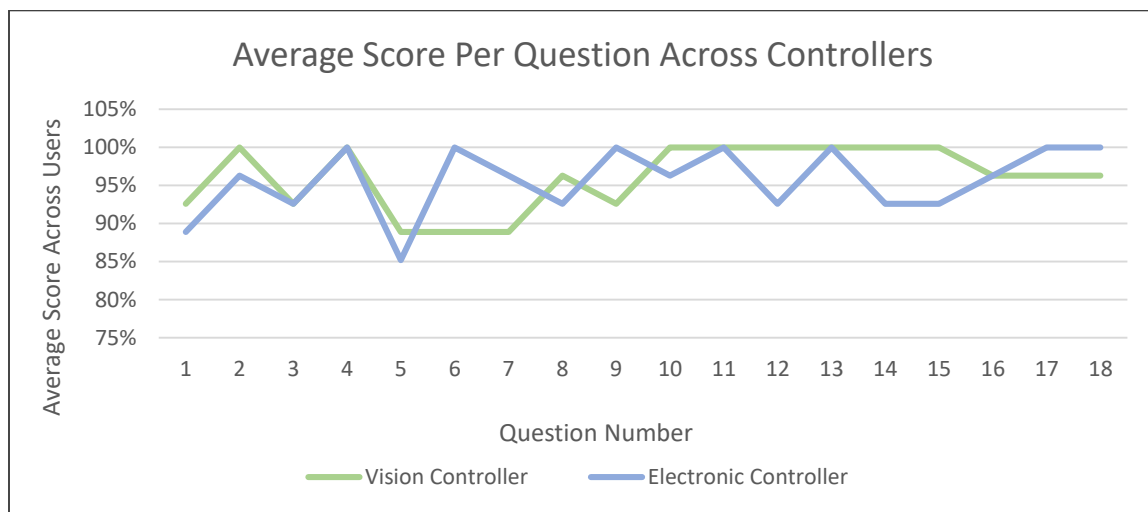


Figure 7.1. Vision controller scores versus electronic controller scores.

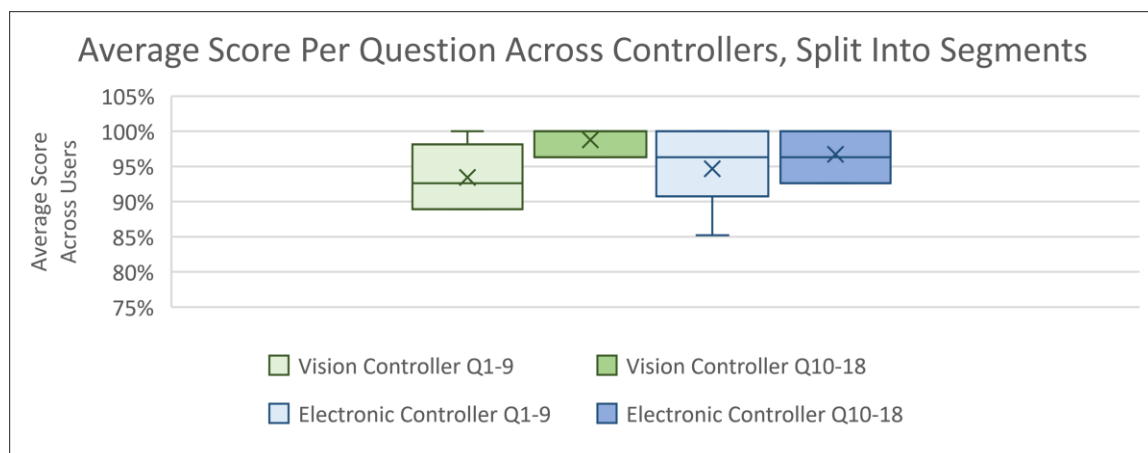


Figure 7.m. Vision controller scores versus electronic controller scores, split into map segments.

The user score per question across controllers can be analysed further by splitting the dataset for each controller into two halves as shown in Figure 7.m. Each half containing 9 questions represents one of the two terrain maps in a given controller's evaluation. It is speculated that users should perform better in the second half due to the learning curve of the controller. Given that this data is a subset of data previously tested, a test for normality is not required. These subsets were compared to each other using the non-parametric WSR Test and the results are summarised in table 7.n. It was shown that users perform statistically better in the second half of the vision controller, suggesting a significant learning curve. In contrast, the electronic controller's performance is statistically the same across the evaluation and this suggests that the controller is intuitive and easy to learn. Figure 7.l. and Figure 7.3.g are also visually indicative that participants perform better in the second half of the vision controller's evaluation when compared to the electronic controller's second half. This would be an interesting result, but as shown in table 7.n, these results have no statistical difference.

Table 7.n. WSR Test Results for Terrain Map Scores across Controllers.

Vision Controller Q1-9 Vs Vision Controller Q10-18	Electronic Controller Q1-9 Vs Electronic Controller Q10-18	Vision Controller Q10-18 Vs Electronic Controller Q10-18
Median = 93% Vs Median = 100%	Median = 96% Vs Median = 96%	Median = 100% Vs Median = 96%
P = 0.017	P = 0.272	P = 0.198
V = 0	V = 3	V = 17
Statistically Different	Statistically Same	Statistically Same

While the value of this subsection's results has been discussed at length, it is important to note that these results are for a game that was not particularly challenging in terms of educational content. This can be seen in the fact that the lowest score for any question was 85% (see Figure 7.l.) Therefore, the conclusions formed in this subsection may change for applications that are more challenging in nature.

7.3.4. Controller Usage

The previous subsections have shown that while the vision controller is more challenging to use, participants perform the same regardless of the controller they use. This suggests that users are able to adapt to challenges and this subsection explores this topic further.

One of the areas where adaptability can be demonstrated is in the way participants make use of magnification as shown in Figure 7.o. Users of the vision controller magnified on average around 15 times for an average of 7 seconds. This means that they spent on average around 105 seconds in the magnification view. In contrast, users of the electronic controller magnified on average around 20 times for about 4 seconds per magnification. This means that these users spent around 80 seconds in the magnification view or around 24% less time than vision controller users. This is an interesting comparison as vision controller magnification can be characterised as detailed inspections, while electronic controller magnifications occur in multiple short bursts. This shows that users naturally adapt to utilise the controller more efficiently. More specifically, because the

magnification toggle is quicker to use on the electronic controller, participants are likely to activate the magnification multiple times if needed. In contrast, because the vision controller is less efficient to use, participants appear to make the most of a single magnification by spending more time in the magnification view. The difference in the number of magnifications could also be explained by the fact that the vision controller supports 6 Degrees of Freedom (DOF) positioning, meaning that it is easier to aim. This would then mean that users of the 3-DOF electronic controller exit the magnified view in order to correct its positioning, resulting in more magnifications. This may explain why users found both controllers to be immersive even though the computer vision controller was found to be more challenging in the GEQ responses. More concretely, the vision controller can be seen as more physically engaging as users utilise its 6 DOF tracking, over resetting the view, to guide magnification.

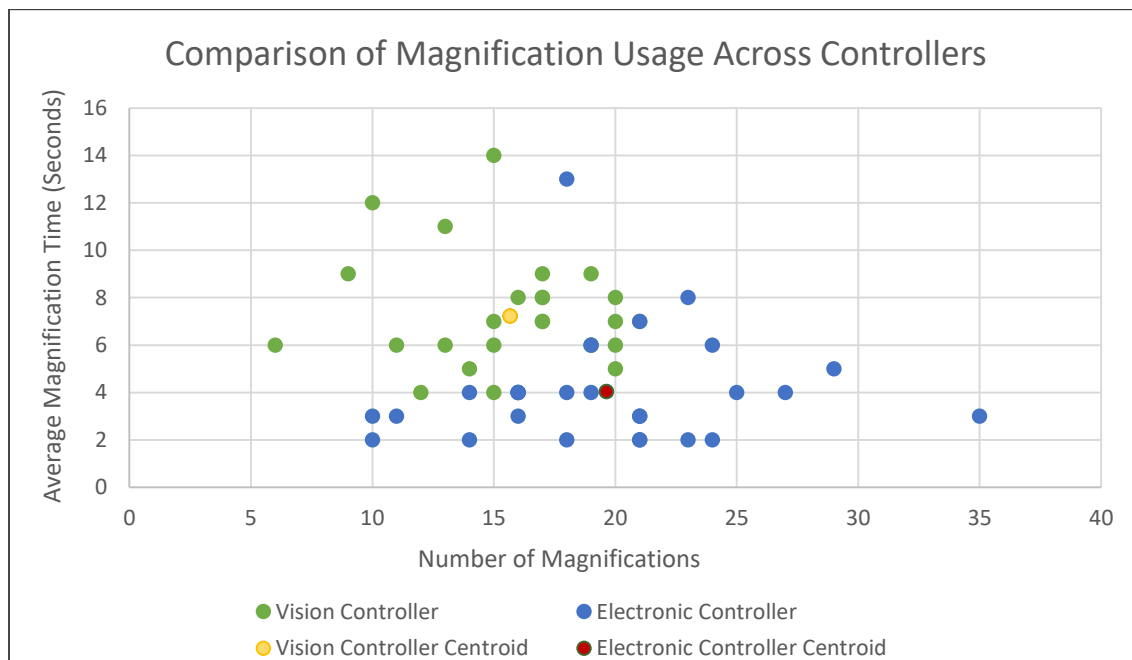


Figure 7.0. A scatter plot showing the way users made use of magnification across controller types. Average centroids have also been included to help with cluster visualization.

7.4. Qualitative Feedback

In addition to the quantitative data collected by the GEQ and the in-game logging, qualitative responses were also gathered from users. This data has been categorised by section, linked with heuristics and is available in Appendix C.

The heuristics chosen for this analysis are the well-established usability heuristics presented by Nielsen [31]. This set of heuristics were chosen over the design principles used in previous sections (i.e., Norman's Design Principles [32]) as they better describe concepts in finer detail. For example, the amount of control a user feels or errors related to a lack of documentation. In other words, the chosen heuristics better differentiate issues relating to a complex system that is focused on usability and immersion.

In addition to the ten heuristics presented by Nielson, ‘immersion’ was added as this enables the identification of feedback relating to aspects that contribute towards immersion, but not to usability. This is important for understanding VR’s usage in an educational setting. For example, feedback relating to the animations of animals does not impact usability, but does contribute to the user’s perception of immersion.

User feedback has been grouped into 3 major sections. These sections are: feedback relating to each controller type and feedback relating to the overall game experience. This feedback was received by users either in written form or verbally and is thus highly subjective, especially for opinions shared by 3 people or less (i.e. 10% of the population).

7.4.1. Electronic Controller Feedback

For the electronic controller, there were 45 positive comments and 33 negative comments and these have been summarised, by heuristic, in Figure 7.p.

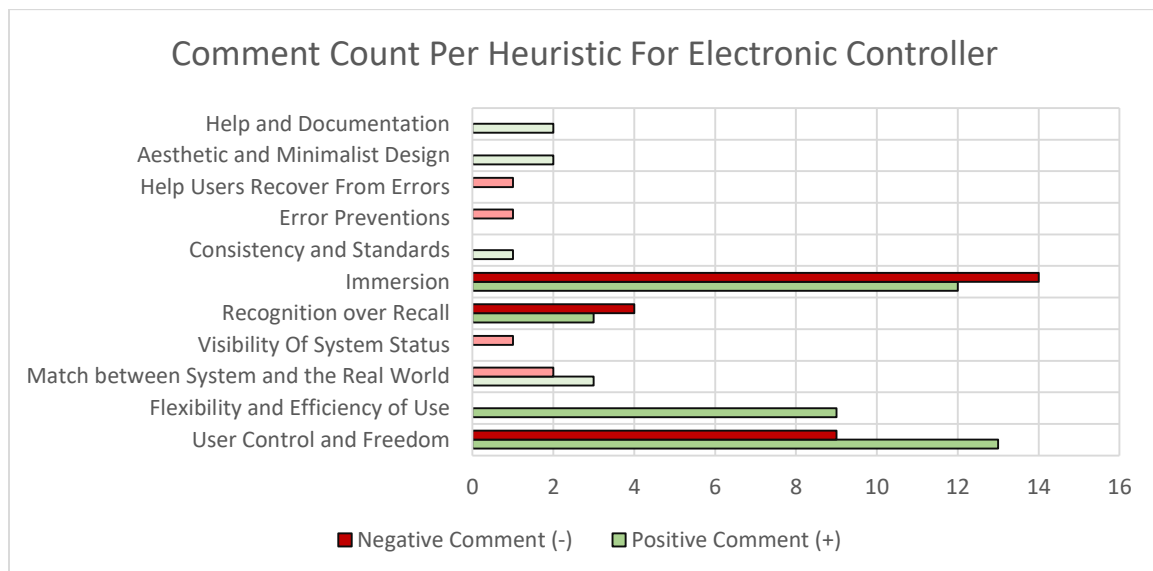


Figure 7.p. Comments from users for the electronic controller grouped by heuristic. Items with 3 or less comments have been faded as these items are highly subjective.

In terms of strengths, it appears that the comments of users emphasised the efficiency (i.e., Flexibility and Efficiency of Use) of the electronic controller, with this category receiving 9 positive comments and no negative comments. The two main elements that contributed to this result were the simplicity of the selection mechanism, which received 5 comments (Appendix C.1-2) and the ergonomics of the controller, which received 3 comments (Appendix C.1-11).

Users also felt that the controller offered them a strong sense of control (i.e., User Control and Freedom) as this category received 13 positive comments. The selection mechanism contributed 8 positive comments to this result as it made users feel in control of their actions (Appendix C.1-1). In contrast, there were also 9 negative comments relating to the User Control heuristic and 6 of these comments were focused around frustration with magnification (Appendix C.1-6). Specifically, it appears that users found that the magnification mechanic was difficult to use, especially when trying to track a flying bird. This is an issue inherent to the controller’s design,

specifically its lack of positional tracking. If the magnification view was controlled using controller's direction instead of the head tracking, a large portion of the central view would move without the user moving their head. This could result in simulator sickness [28]. However, it should be noted that 5 users were explicitly happy with the implementation of the magnification mechanic (Appendix C.1-4).

The intuitiveness (i.e. recognition over recall) of the controller was also discussed in the user feedback. 3 users felt that the controller had a design familiar to them (Appendix C.1-10), while 4 users kept mixing up the selection and magnification toggle buttons (Appendix C.1-13). This is a challenge in creating a generic controller: it is familiar across applications, but not specialised for any of them. This means that controls are easy to pick up if a user has prior experience with a similar controller, but potentially also easy to forget.

The last point of contention in the user feedback was the topic of immersion, with 14 negative comments and 12 positive comments. The main factors driving negative feedback were 5 comments about the controller not being positionally tracked (Appendix C.1-12), 3 comments about connectivity issues disrupting the experience (Appendix C.1-14) and 5 participants feeling that the efficiency of the controller disrupts the enjoyment of the game (Appendix C.1-25, C.1-26). In contrast, 9 participants felt that they were positively unaware of their surroundings (Appendix C.1-23), with 3 users specifically not feeling time pressured (Appendix C.1-24). This suggests that the electronic controller contributes to immersion by providing an accurate and efficient means of controlling the game environment. It is in a sense invisible to the user as they do not have to think about how it is used, this enhances a user's sense of skill.

7.4.2. Vision Controller Feedback

The vision controller received 45 positive comments and 61 negative comments and these have been summarised by heuristic in Figure 7.q.

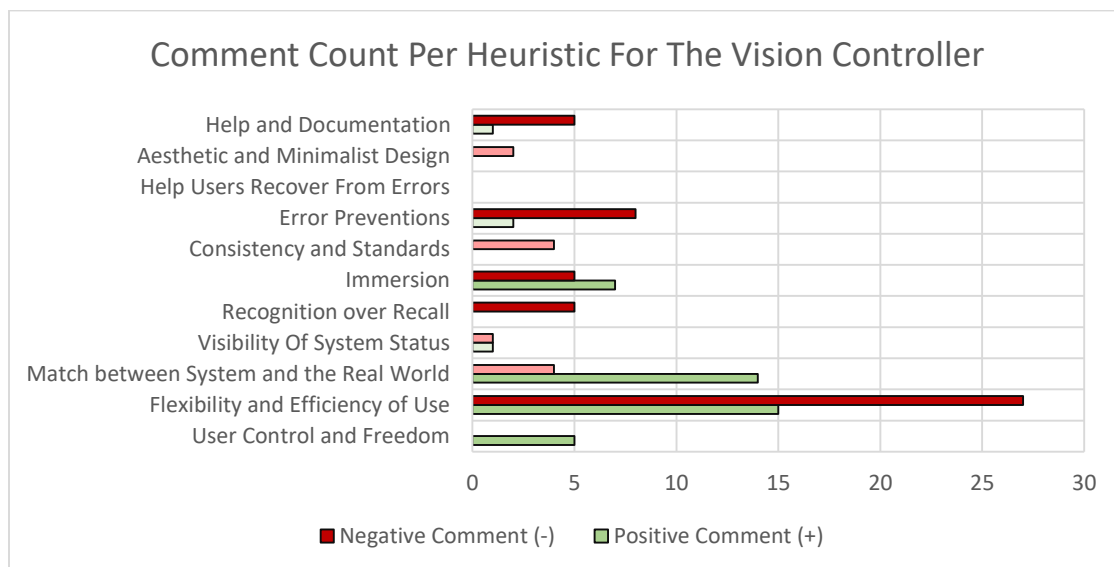


Figure 7.q. Comments from users for the vision controller grouped by heuristic. Items with around 3 or less comments have been faded as these items are highly subjective.

As seen in the section on score performance (section 7.3.3), vision controller users scored higher in the second half of their evaluations. This result suggests that the controller is less intuitive to use and requires a training stage. This hypothesis appears to be supported in feedback regarding Help and Documentation as 3 users reported that they did not understand the way the gear worked (Appendix C.2-28), 1 user reported not initially understanding that the controller must be in the line of sight of the camera (Appendix C.2-29) and another user did not initially know where the questions were displayed (Appendix C.2-30). It is expected that more users experienced this problem and did not explicitly report it. This issue could be remedied in an application scenario by having a tutorial stage that is aimed at teaching users how to use the controller. This was not possible in this study as this would likely add to the issue of fatigue and increase the likelihood of participants experiencing simulator sickness due to prolonged exposure to VR.

Another problematic area for the vision controller was in its ability to prevent errors, with 8 comments relating negatively to this topic. 4 users reported having difficulty aiming the magnification view (Appendix C.2-10) and another 4 users reported being unable to determine if issues with selection were because of them or because of tracking issues (Appendix C.2-15). This feedback is important to consider given that these issues likely result in users feeling like they are less in control, which will negatively impact their experience.

For the Immersion heuristic, 7 participants reported feeling positively immersed in the experience (Appendix C.2-31), while 5 participants felt that the learning curve of the controller negatively impacted their experience (Appendix C.2-33). This feedback suggests that the controller is capable of positively contributing to immersion as confirmed by the GEQ, though users need more time and scaffolding to be able to be comfortable with the vision controller. This idea is further supported by the 4 negative comments related to the heuristic, recognition over recall. Specifically, 2 users forgot about the magnification feature (Appendix C.2-11) and 2 users forgot how to submit answers (Appendix C.2-6). These issues suggest that the vision controller's experience had poor affordance, which could be remedied through prior training. Additionally, the interface could be upgraded to show more hints about the next expected action.

The area where the most negative feedback was received was in the area of flexibility and efficiency of use. This result was expected as the GEQ proved that the vision controller is a more challenging controller to use. Additionally, the metric analysis conducted in the previous chapter also demonstrated that the vision controller takes more time to use. 8 users felt that the submission box was too far away (Appendix C.2-3), 15 users felt that the gear took too long to rotate before a change in selection was registered (Appendix C.2-14) and 4 users had issues submitting answers due to issues related to line of sight (Appendix C.2-4, C.2-5). In contrast to this, 15 users enjoyed the magnification mechanic (Appendix C.2-8), with 4 of these users believing that this magnification approach was superior (Appendix C.2-9). This suggests that users felt empowered by the magnification mechanic and burdened by the selection and submission mechanics. This contrasts with the results of the electronic controller in the previous subsection, where users were frustrated by the magnification mechanic and praised the efficiency of the selection and submission mechanics.

The last heuristic to consider in Figure 7.q is the match between system and the real world. This appears to be the heuristic that performed the best for the vision controller as there are 14 positive comments and only 4 negative comments. Positive comments were related to the intuitiveness of the controls (Appendix C.2-1, C.2-9), the ergonomics of the controller (appendix C.2-13) and unexpected conveniences (selection wrap around, selection persistence during magnification) of

the controller (Appendix C.2-2, C.2-19). It should be noted that 4 participants did feel that the controller felt jittery in certain circumstances (Appendix C.2-16) which contributed negatively to this heuristic. This suggests that the controller can positively contribute towards immersion by providing a physically engaging interface, though aspects of this interface, such as the selection scrolling and submission box, could be improved.

7.4.3. Game Experience Feedback

While the educational game was not the main focus of this study, it received 49 positive comments and 40 negative comments. This data is summarised in Figure 7.r.

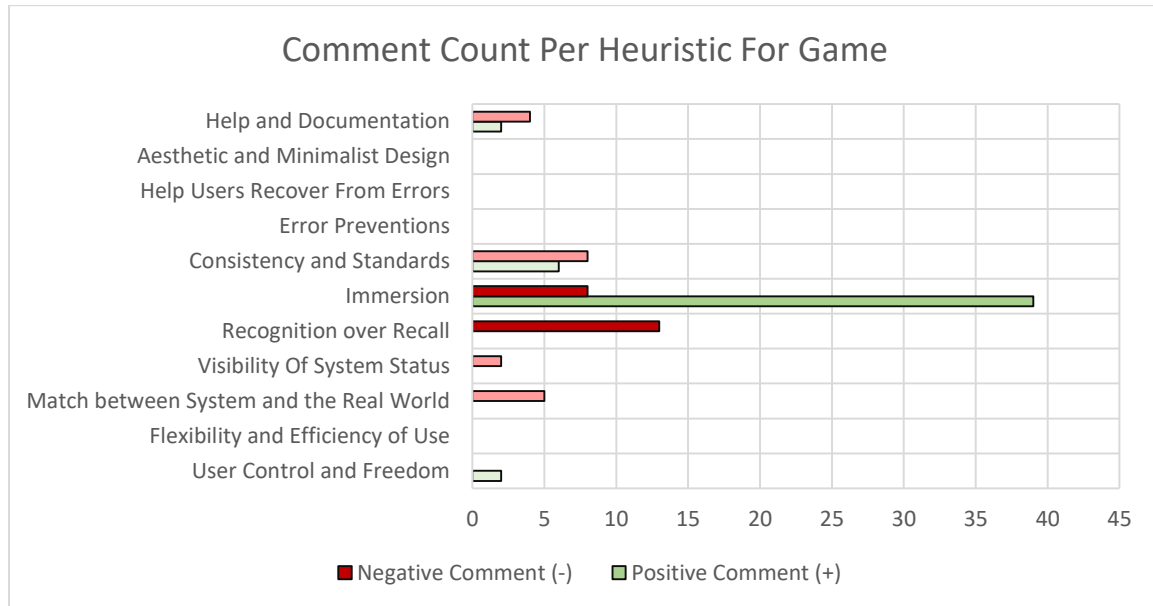


Figure 7.r. Comments from users for the game experience grouped by heuristic. Items with around 3 or less comments have been faded as these items are highly subjective.

In looking over the feedback for the game, two main insights emerge. Firstly, it appears that the game had a positive effect on immersion as there were 39 positive comments related to this. Comments ranged from enjoying the overall experience (Appendix C.3-6, C.3-7, C.3-24, C.3-25) to enjoying specific artistic elements of the game (Appendix C.3-16, C.3-17, C.3-18, C.3-19, C.3-20). This adds to the narrative that virtual reality is useful for improving immersion in an educational context. That being said, there were 3 users that reported feeling eye strain (Appendix C.3-27) and issues like this need to be considered when deciding to use VR for educational applications.

The second insight was that a critical concept of the educational experience was initially misunderstood by a large number of participants. Specifically, 13 participants did not understand the concept of areas (Appendix C.3-2) even though it was explained in the introduction videos and during an initial in-game voice prompt. This is problematic as this concept is important for answering the multiple choice questions. This negatively impacted the heuristic recognition over recall as users were unable to recognise what this concept meant. This is an issue that can be remedied with a training stage or a longer introduction video. It is speculated that this issue occurred because users were overloaded by the many new concepts presented in this experiment and this idea is supported by related feedback given in Appendix C.3-4.

7.5. Conclusions

An evaluation was carried out with 27 participants to determine whether a computer-vision based controller can be as immersive as a conventional electronic controller. Processing the quantitative and qualitative data yielded the following findings:

1. The vision controller performs approximately equivalently to the electronic controller as they both achieved a high immersion score (median of 3.4 out of 4.0) in the GEQ. They are also statistically the same in every other GEQ metric other than the challenge metric. This means that they are both adequate controllers for educational virtual reality experiences and that the vision controller is more challenging to use in comparison to the electronic controller.
2. It was found that a number of attributes contributed to the vision controller being perceived as more challenging. The scrolling of the selection gear took too long to register a change and the submission box was considered to be too far away by some participants. The intuitiveness of the controller's selection mechanic along with the immersiveness of the magnification mechanism were redeeming factors.
3. Both controllers contributed positively to immersion, but they did so in different ways. The electronic controller was efficient and easy to use, which made participants feel in control, while the vision controller was physically interactive which made participants feel engaged.
4. The vision controller required longer to learn based on the quantitative data collected. To insure that this does not impact user performance, a dedicated learning stage should precede any assessments to ensure that users are comfortable with the controller. Failing to do so may result in users not understanding key concepts of the assessment.

Based on the findings presented above, it can be concluded that a computer-vision based virtual reality controller can provide comparable immersion to a conventional electronic controller, even though the vision controller has no internal electronics.

Chapter 8:

Conclusions & Future Work

8.1. Conclusions

Mobile virtual reality has the potential to improve learning experiences by making them more immersive and engaging for students. This type of virtual reality also aims to be more cost effective by using a smartphone to drive the virtual reality experience. One issue with mobile virtual reality is that the screen (i.e., the main interface) of the smartphone is occluded by the virtual reality headset.

To investigate solutions to this issue, this project details the development and testing of a computer-vision based controller that aims to have a cheaper per unit cost than a conventional electronic controller. Reducing the cost per unit is useful in an educational context as solutions would need to scale to the typical number of pupils in a classroom. The research question for this project is thus, *can a computer-vision based virtual reality controller provide comparable immersion to a conventional electronic controller?*

The vision controller was developed as two separate systems and the following conclusions were formed during the course of this development:

- **Backend Computer Vision System** - A feature matching based computer vision system is more robust, but slower than a system that works with fiducial markers. In order to have a balance between Robustness and Speed, a feature matching system should make use of ORB over SIFT or SURF, as ORB, based on our experiments is 10 times faster than SIFT (the most accurate of the three) with only a 6% loss in accuracy (82% vs 88%). Lastly, this solution can be sped up by around 2 times by using the Lucas-Kanade optical flow method, which leverages past detections to speed up feature detection in frames with coherent motion changes.
- **Frontend Virtual Reality Interface** – An interface was developed around the computer vision system using a user centred design process. The first phase made use of a low fidelity paper prototype to test out interface mechanisms and it was found that having a more unified interface (i.e., fewer separate props) was necessary as manipulating multiple physical objects is difficult when using a head-mounted display. The second phase used an iterative design process to develop a 3D printable controller and matching 2D printable paper trackers. The main per unit cost of the controller is around \$3, an order of magnitude cheaper than the Gear VR controller (\$34). It should be noted that the 3D printer used costs around \$350, though this is a one-time fixed cost. The final phase was to develop the software components of the interface, which included its rendering in VR and scaffolding around its use (e.g., showing hints if the controller was too far away).

An educational VR game was developed to test the computer-vision based interface against a conventional electronic controller (the Samsung Gear VR Controller). This project's research question was evaluated using quantitative data from the Game Experience Questionnaire [17] and performance metrics, and qualitative data in the form of feedback generated by the 27 participants.

The evaluation process revealed that the vision controller provides an approximately equivalent degree of immersion to the electronic controller. The vision controller was also perceived to be more challenging to use, though this challenge contributed to a user's sense of flow rather than disturbing their educational experience. This conclusion arose from the findings that the electronic controller is an efficient controller that gives users a sense of control, while the vision controller provides a physically interactive experience that makes users feel engaged.

These findings allow for the answering of the project's research question, specifically, *a computer-vision based virtual reality controller can provide comparable immersion to a conventional electronic controller*. This is possible even though the vision controller does not make use of any dedicated electronics and has a cheaper per unit cost (\$3 vs \$34). This cost reduction means that a computer-vision based solution can be deployed to more classrooms for the same price as a solution based around the Samsung GearVR controller. Also, since the vision controller can be fabricated using a 3D PLA printer and a 2D ink printer, it can be used in scenarios where electronic controllers are not available for purchase.

8.2. Future Work

There are three main areas where the vision controller could be improved:

1. The most problematic aspect of the vision controller was the amount of time between gear rotations and changes in selection being registered in game. As explained in previous chapters, the current selection gear was designed as an optimisation of two opposing factors. Specifically, the selection intervals on the selection gear should be large enough to improve tracking reliability, but small enough to reduce the amount of scrolling needed between intervals. A multi-gear system could be developed to create a controller with good reliability and faster scroll times, though it should be noted that fast moving markers have a reduced tracking reliability due to motion blur.
2. Another flaw of the vision controller interface was the placement of the submission boxes. Submission boxes need to be placed out of the way of the user so that accidental submissions do not occur, but they cannot be so far that users become frustrated. It is likely that optimal submission box placement may differ across participants, which means that either users should be able to place submission boxes or an automated method needs to be developed for placing them. Alternatively, research could also be done into alternative ways of submitting answers.
3. During the course of the evaluation it was noted that participants do not always remember to face the controller towards the phone's camera. This results in reduced tracking fidelity and controller jitter. Research could be done into controller shapes that have more robust viewing angles. An alternate solution to this problem could also be found in the use of mirrors and lenses for improving the field of view of a given smartphone's camera.
4. Alternate controller designs and educational applications should be evaluated to further generalize the idea of a computer-vision controller. Controller designs could vary in shape and number of components, and applications could vary in content and difficulty. Applications could also be tested in actual classrooms to measure knowledge retention.

References

- [1] Amin, A., Gromala, D., Tong, X., Shaw, C., “Immersion in Cardboard VR Compared to a Traditional Head-Mounted Display,” in *International Conference on Virtual, Augmented and Mixed Reality*, pp. 269–276, 2016.
- [2] Bay, H., Tuytelaars, T., Van Gool, L., “Speeded-Up Robust Features (SURF),” *European conference on computer vision*, pp. 404–417, 2006.
- [3] Bellotti, F., Kapralos, B., Lee, K., Moreno-Ger, P., Berta, R., “Assessment in and of Serious Games: An Overview,” *Advances in Human-Computer Interaction*, pp. 1–11, 2013.
- [4] Bharambe, A., Douceur, J. R., Lorch, J. R., Moscibroda, T., Pang, J., Seshan, S., Zhuang, X., “Donnybrook: Enabling Large-Scale, High-Speed, Peer-to-Peer Games,” *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, p. 389, 2008.
- [5] Bradski, G., “Computer Vision Face Tracking for Use in a Perceptual User Interface,” 1998, [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.7673>.
- [6] Brockmyer, J., Fox, C., Curtiss, K., McBroom, E., Burkhart, K., Pidruzny, J., “The Development of the Game Engagement Questionnaire: A Measure of Engagement in Video Game-Playing,” *Journal of experimental social psychology*, vol. 45, no. 4, pp. 624–634, 2009.
- [7] Calonder, M., Lepetit, V., Strecha, C., Fua, P., “BRIEF: Binary Robust Independent Elementary Features,” in *European conference on computer vision*, pp. 778–792, 2010.
- [8] Chatzilari, E., Liaros, G., Nikolopoulos, S., Kompatsiaris, Y., “A Comparative Study on Mobile Visual Recognition,” in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp. 442–457, 2013.
- [9] Comaniciu, D., Meer, P., “Mean Shift: A Robust Approach Toward Feature Space Analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.
- [10] Cunningham, D., Wallraven, C., *Experimental design: From user studies to psychophysics*. CRC Press, 2011.
- [11] Denisova, A., Nordin, A. I., Cairns, P., “The Convergence of Player Experience Questionnaires,” *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, pp. 33–37, 2016.
- [12] Dicheva, D., Dichev, C., Agre, G., Angelova, G., “Gamification in Education: A Systematic Mapping Study,” *Journal of Educational Technology & Society*, vol. 18, no. 3, p. 75, 2015.
- [13] Fiala, M., “Designing Highly Reliable Fiducial Markers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1317–1324, 2010.
- [14] Fischler, M. A., Bolles, R. C., “Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [15] Gugenheimer, J., Dobbstein, D., Winkler, C., Haas, G., Rukzio, E., “FaceTouch: Enabling Touch Interaction in Display Fixed UIs for Mobile Virtual Reality,” *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 2016.

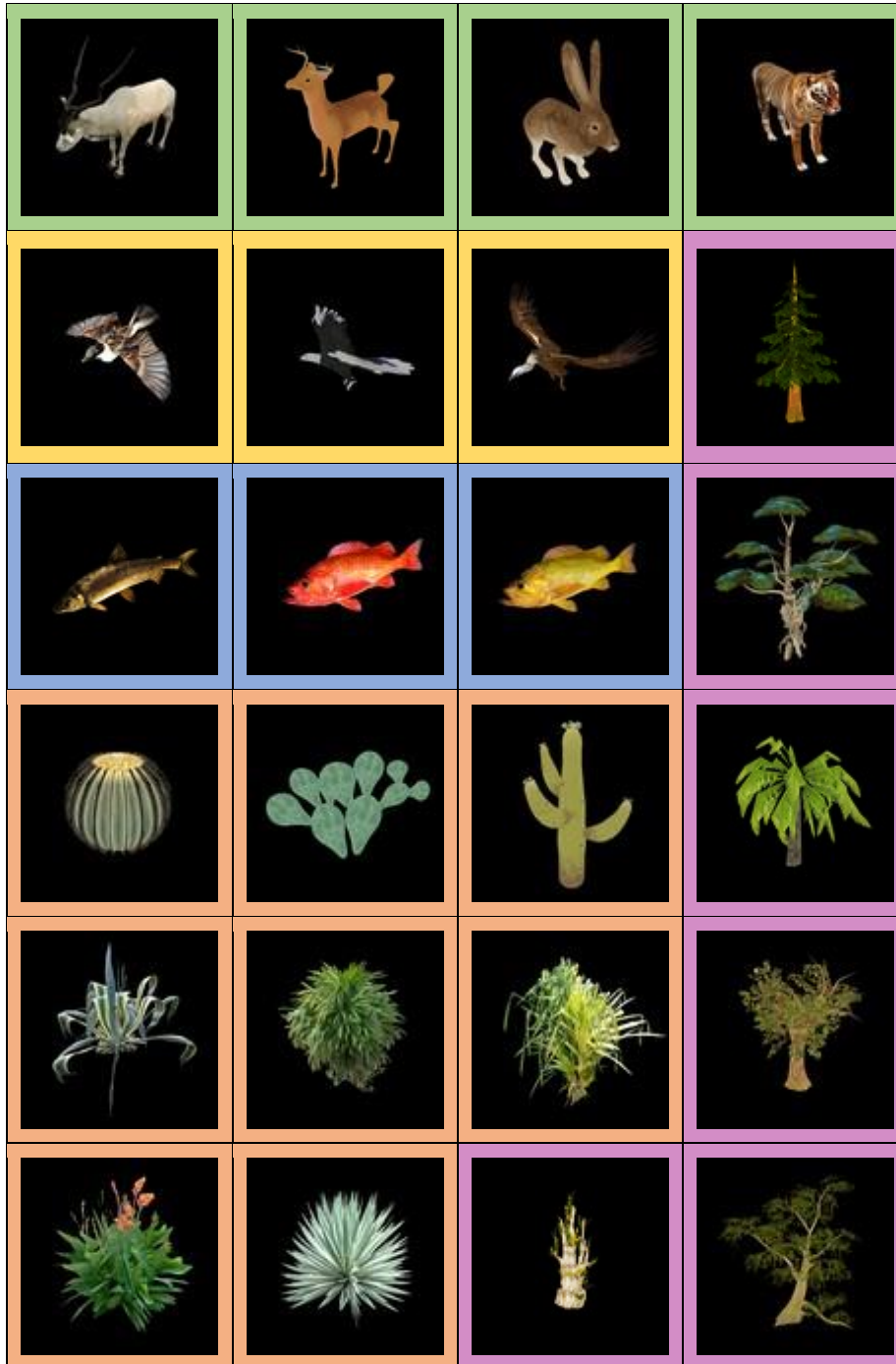
- [16] Harris, C., Stephens, M., "A Combined Corner and Edge Detector," *Alvey vision conference*, vol. 15, 1988.
- [17] IJsselsteijn, W., De Kort, Y., Poels, K., "The Game Experience Questionnaire," *Eindhoven: Technische Universiteit Eindhoven.*, 2013. .
- [18] Jennett, C., Cox, A., Cairns, P., Dhoparee, S., Epps, A., Tijs, T., Walton, A., "Measuring and Defining the Experience of Immersion in Games," *International journal of human-computer studies*, vol. 66, no. 9, pp. 641–661, 2008.
- [19] Jin, S. A. A., "Toward Integrative Models of Flow: Effects of Performance, Skill, Challenge, Playfulness, and Presence on Flow in Video Games," *Journal of Broadcasting and Electronic Media*, vol. 56, no. 2, pp. 169–186, 2012.
- [20] Kato, K., Miyashita, H., "Creating a Mobile Head-mounted Display with Proprietary Controllers for Interactive Virtual Reality Content," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pp. 35–36, 2015.
- [21] Krekhov, A., Emmerich, K., Bergmann, P., Cmentowski, S., Krüger, J., "Self-Transforming Controllers for Virtual Reality First Person Shooters," *Proceedings of the Annual Symposium on Computer-Human Interaction in Play.*, pp. 517–529, 2017.
- [22] Kroes, M., Ament, T., "A Low-cost Tracking Solution For VR Headsets," 2017, [Online]. Available: <https://repository.tudelft.nl/islandora/object/uuid:28188648-0324-436d-8b63-700442963d24>.
- [23] Lowe, D. G., "Object Recognition From Local Scale-Invariant Features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pp. 1150–1157 vol.2, 1999.
- [24] Lowe, D. G., "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [25] Lucas, B., Kanade, T., "An Iterative Image Registration Technique With an Application to Stereo Vision," 1981, [Online]. Available: https://ri.cmu.edu/pub_files/pub3/lucas_bruce_d_1981_2/lucas_bruce_d_1981_2.pdf.
- [26] Lugrin, J. L., Ertl, M., Krop, P., Klupfel, R., Stierstorfer, S., Weisz, B., Ruck, M., Schmitt, J., Schmidt, N., Latoschik, M. E., "Any 'Body' There? Avatar Visibility Effects in a Virtual Reality Game," in *25th IEEE Conference on Virtual Reality and 3D User Interfaces, VR*, pp. 17–24, 2018.
- [27] Lv, Z., Halawani, A., Feng, S., ur Réhman, S., Li, H., "Touch-Less Interactive Augmented Reality Game on Vision-Based Wearable Device," *Personal and Ubiquitous Computing*, vol. 19, no. 3–4, pp. 551–567, 2015.
- [28] Moss, J. D., Muth, E. R., "Characteristics of Head-Mounted Displays and Their Effects on Simulator Sickness," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 53, no. 3, pp. 308–319, 2011.
- [29] Moss, J. D., Muth, E. R., Tyrrell, R. A., Stephens, B. R., "Perceptual Thresholds for Display Lag in a Real Visual Environment Are Not Affected by Field of View or Psychophysical Technique," *Displays*, vol. 31, no. 3, pp. 143–149, 2010.
- [30] Nichols, S., Patel, H., "Health and Safety Implications of Virtual Reality: A Review of Empirical Evidence," *Applied Ergonomics*, vol. 33, no. 3, pp. 251–271, 2002.

- [31] Nielsen, J., "Enhancing the Explanatory Power of Usability Heuristics," in *Conference on Human Factors in Computing Systems*, pp. 152–158, 1994.
- [32] Norman, D., *The Design of Everyday Things (Revised and Expanded Edition)*. 2013.
- [33] Norman, D., *The psychology of everyday things*. Basic Books, 1988.
- [34] Norman, D., Draper, S., "User Centered System Design; New Perspectives on Human-Computer Interaction," 1986, [Online]. Available: <https://dl.acm.org/citation.cfm?id=576915%C3%DC>.
- [35] Pelargos, P. E., Nagasawa, D. T., Lagman, C., Tenn, S., Demos, J. V., Lee, S. J., Bui, T. T., Barnette, N. E., Bhatt, N. S., Ung, N., Bari, A., Martin, N. A., Yang, I., "Utilizing Virtual and Augmented Reality for Educational and Clinical Enhancements in Neurosurgery," *Journal of Clinical Neuroscience*, vol. 35, pp. 1–4, 2017.
- [36] Powell, W., Powell, V., Brown, P., Cook, M., Uddin, J., "Getting Around in Google Cardboard – Exploring Navigation Preferences With Low-Cost Mobile VR," in *2016 IEEE 2nd Workshop on Everyday Virtual Reality (WEVR)*, pp. 5–8, 2016.
- [37] Regan, C., "An Investigation Into Nausea and Other Side-Effects of Head-Coupled Immersive Virtual Reality," *Virtual Reality*, vol. 1, no. 1, pp. 17–31, 1995.
- [38] Rosten, E., Drummond, T., "Machine Learning for High-Speed Corner Detection," in *European Conference on Computer Vision*, pp. 430–443, 2006.
- [39] Rublee, E., Rabaud, V., Konolige, K., Bradski, G., "Orb: An Efficient Alternative to Sift or Surf," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2564–2571, 2011.
- [40] Ryan, R. M., Rigby, C. S., Przybylski, A., "The Motivational Pull of Video Games: A Self-Determination Theory Approach," *Motivation and Emotion*, vol. 30, no. 4, pp. 347–363, 2006.
- [41] Sharma, P., Prashant, "Challenges With Virtual Reality on Mobile Devices," in *ACM SIGGRAPH*, pp. 1–1, 2015.
- [42] Shute, V. J., Ventura, M., Bauer, M., Zapata-Rivera, D., "Melding the Power of Serious Games and Embedded Assessment to Monitor and Foster Learning: Flow and Grow," in *Serious Games: Mechanisms and Effects*, 2009.
- [43] Sibert, L. E., Jacob, R. J. K., "Evaluation of Eye Gaze Interaction," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 281–288, 2000.
- [44] Smus, B., Riederer, C., "Magnetic Input for Mobile Virtual Reality," in *Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pp. 43–44, 2015.
- [45] Sutherland, I. E., "A Head-Mounted Three Dimensional Display," in *Proceedings of the joint computer conference*, p. 757, 1968.
- [46] Swain, M. J., Ballard, D. H., "Indexing via Color Histograms," in *Active perception and robot vision.*, pp. 261–273, 1992.
- [47] Tareen, S. A. K., Saleem, Z., "A Comparative Analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," in *International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pp. 1–10, 2018.

- [48] Tregillus, S., Folmer, E., “VR-STEP: Walking-in-Place using Inertial Sensing for Hands Free Navigation in Mobile VR Environments,” *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016.
- [49] Tregillus, S., Al Zayer, M., Folmer, E., “Handsfree Omnidirectional VR Navigation using Head Tilt,” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 4063–4068, 2017.
- [50] Woolson, R. F., *Wilcoxon Signed-Rank Test*. John Wiley & Sons, Inc., 2007.
- [51] Yan, X., Fu, C.-W., Mohan, P., Goh, W. B., “CardboardSense: Interacting with DIY Cardboard VR Headset by Tapping,” in *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, pp. 229–233, 2016.
- [52] Yoo, S., Kay, J., “VRun,” in *Proceedings of the 28th Australian Conference on Computer-Human Interaction*, pp. 562–566, 2016.
- [53] Yoo, S., Parker, C., “Controller-less Interaction Methods for Google Cardboard,” in *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, pp. 127–127, 2015.
- [54] Zikas, P., Bachlitzanakis, V., Papaefthymiou, M., Papagiannakis, G., “A Mobile, AR Inside-Out Positional Tracking Algorithm, (MARIOPOT), Suitable for Modern, Affordable Cardboard-Style VR HMDs,” in *Euro-Mediterranean Conference*, pp. 257–268, 2016.

Appendix A: Educational Game Assets


A.1. Plants and Animal Assets



Appendix A.1. Plants and Animals included in the Educational Game. Green tiles are land animals, yellow tiles are birds (exclusive to sky), blue tiles are fish (exclusive to water areas), purple tiles are trees and orange tiles are small plants.

Appendix B: Evaluation Documentation

B.1. Participant Consent Form

DEPARTMENT OF COMPUTER SCIENCE		
UNIVERSITY OF CAPE TOWN	RESEARCHER/S: Tashiv Sewpersad	
PRIVATE BAG X3	TELEPHONE: +27-73-919 5735	
RONDEBOSCH 7701	E-MAIL: Swptas001@myuct.ac.za	
SOUTH AFRICA	URL: https://www.cs.uct.ac.za	

Informed Voluntary Consent to Participate in Research Study

Project Title: A Low Cost Virtual Reality User Interface for Educational Games

Invitation to participate, and benefits: You are invited to participate in a research study conducted with Virtual Reality equipment. The study aims to investigate the immersion levels of two different physical interfaces in the context of a smartphone-driven educational game. I believe that your experience would be a valuable source of information, and hope that by participating you may gain useful knowledge.

Procedures: During this study, you will be asked to play a plant and animal identification game using a virtual reality headset and (at different times) the interfaces being considered. You will also be asked to perform performance and usability evaluations in the form of a questionnaire.

Recording: The choices you make in the game will be recorded as part of the performance evaluation. Identifying information will not be present in the data collected.

Risks: Some users of virtual reality equipment experience temporary symptoms of nausea. You are free to withdraw from this study, should you experience any discomfort during the experiment. More information regarding this procedure is given below.

Disclaimer/Withdrawal: Your participation is completely voluntary; you may refuse to participate, and you may withdraw at any time without having to state a reason and without any prejudice or penalty against you. Should you choose to withdraw, the researcher commits not to use any of the information you have provided without your signed consent. Note that the researcher may also withdraw you from the study at any time.

Confidentiality: All information collected in this study will be kept private in that you will not be identified by name or by affiliation to an institution. Confidentiality and anonymity will be maintained as pseudonyms will be used.

What signing this form means: By signing this consent form, you agree to participate in this research study. The aim, procedures to be used, as well as the potential risks and benefits of your participation have been explained verbally to you in detail, using this form. Refusal to participate in or withdrawal from this study at any time will have no effect on you in any way. You are free to contact me, to ask questions or request further information, at any time during this research.

I agree to participate in this research (tick one box) ☐ Yes ☐ No _____ (Initials)

_____ Name of Participant	_____ Signature of Participant
_____ Name of Researcher	_____ Signature of Researcher

Approved by the Science Faculty Research Ethics Committee, 3 May 2019

Appendix B.1. A form that participants sign before taking part in the user experiment described in chapter 6. This form ensures that the consent provided by participants is informed as key details of the experiment are included in it.

B.2. Pre-Experiment Questionnaire

DEPARTMENT OF Computer science

UNIVERSITY OF CAPE TOWN
PRIVATE BAG X3
RONDEBOSCH 7701
SOUTH AFRICA

RESEARCHER: Tashiv Sewpersad
TELEPHONE: +27-21-650 2663
E-MAIL: swptas001@myuct.ac.za
URL: <https://www.cs.uct.ac.za>



Participant Information Form

Participant Number: _____

Experiment Order:

	EL		
Order 1			
Order 2			
	CV		

Age: _____

Gender: _____

Field of Study: _____

How often do you play 3D Games:

<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Never	At most once a month	At most once a week	At most once a day

Previous Experience with VR Games:

<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
None	Tried once	Tried a few times	Use frequently

Previous experience with educational games:

<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
None	Tried once	Tried a few times	Use frequently

Appendix B.2. A form that is completed before the experiment described in chapter 6 takes place. It records potential confounding factors as well as the experiment's configuration.

B.3. User Evaluation Questionnaire (2 Pages)

Participant ID: _____

A. How did you feel DURING the game?

ID	Question	Not At All	Slightly	Moderately	Fairly	Extremely
0	<i>E.g. Answer like this</i>			x		
1	I felt content					
2	I felt skilful					
4	I thought it was fun					
5	I was fully occupied with the game					
6	I felt happy					
7	It gave me a bad mood					
8	I thought about other things					
9	I found it tiresome					
10	I felt competent					
11	I thought it was hard					
12	It was aesthetically pleasing					
13	I forgot everything around me					
14	I felt good					
15	I was good at it					
16	I felt bored					
17	I felt successful					
18	I felt imaginative					
19	I felt that I could explore things					
20	I enjoyed it					
21	I was fast at reaching the game's targets					
22	I felt annoyed					
23	I felt pressured					
24	I felt irritable					
25	I lost track of time					
26	I felt challenged					
27	I found it impressive					
28	I was deeply concentrated in the game					
29	I felt frustrated					
30	It felt like a rich experience					
31	I lost connection with the outside world					
32	I felt time pressure					
33	I had to put a lot of effort into it					

Page 1 of 2

B. Any NEGATIVE feedback about your experience?

C. Any POSITIVE feedback about your experience?

Page 2 of 2

Appendix B.3. These two pages are provided to users after each of the two interfaces are evaluated. The first page is the core module of the GEQ proposed by IJsselsteijn et al. The second page provides users with the opportunity to provide further feedback about their overall experience.

B.4. User Orientation Slides (2 Pages)

A Low Cost Virtual Reality Interface for Educational Games

Interface Evaluation

The Study

- **Aim:** Investigate immersion levels of two different controllers.



- **Context:** Smartphone-driven virtual reality educational game.

- **VR Note:** If you experience any symptoms of simulation sickness, please notify me immediately.

The Game

- You are asked to identify plants/animals on areas of the terrain. There is an **area guide**.
- There are **2 terrains** per controller and each terrain has **9 questions** each.
- Each question has **3 options** to choose from.



Electronic Controller Interface



Electronic Controller

Usage Steps:

1. **Point** at **answer** on board in front of you.
2. Press **TRIGGER** to select answer.



Controller Evaluation Time (15 Minutes)



Questionnaire Time (10 Minutes)



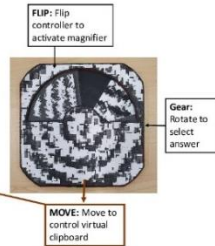
Computer Vision
Controller Interface



Computer Vision Controller

• Usage Steps:

1. Look at the controller.
2. Rotate GEAR to select answer.
3. Tap SUBMISSION BOX to submit answer.



Controller Evaluation Time (15 Minutes)



Questionnaire Time (10 Minutes)



Evaluation Complete :)



Appendix B.4. These 12 slides aim to orientate the users about the research project, the aim of the evaluation software and the way each interface should be used. Users are shown the relevant slides during different stages of the evaluation process.

B.5. Instructional Documentation: Procedure Guide

Your Participant ID: _____

Procedure Guide

*Please follow these steps in order to complete the study, call me (Tashiv) on [REDACTED] for queries.
Also remember that as a user in this experiment, your feedback is always valid.*

Step 1: Intro

- Remove the smartphone from the VR headset by following this guide. →
- On the smartphone, press the power button (top right side). Swipe up to unlock. You are now on the home screen.
- Watch the *Intro* video in the **Step 1** section on home screen, by pressing its icon.

Step 2: Test A

- Watch the *Intro* video in the **Step 2** section on home screen, by pressing its icon.
- Play the game (i.e. Test A) by pressing on the *Game* icon in the **Step 2** section on home screen. Follow the on-screen instructions until the game is completed.
- Fill in the two sided questionnaire for Test A. →

Step 3: Test B

- Watch the *Intro* video in the **Step 3** section on home screen, by pressing its icon.
- Play the game (i.e. Test B) by pressing on the *Game* icon in the **Step 3** section on home screen. Follow the on-screen instructions until the game is completed.
- Fill in the two sided questionnaire for Test B. →

Terminology


Smartphone


VR Headset


Electronic Controller


Vision Controller

Appendix B.5. This guide is supplied to users so that they can conduct an evaluation on their own remotely. The colour coding matches the home-screen of the device they are testing with.

B.6. Instructional Documentation: Headset Operation Guide

Steps to Remove Smartphone from Virtual Reality Headset

1. Pull the latch away from the smartphone



2. Swing the smartphone away from the headset



3. Pull the smartphone away from and off the connector hinge to release it



(Run these steps in reverse to insert smartphone into virtual reality headset)

Appendix B.6. Users are given the VR headset with the smartphone already docked in it. This is done to protect the smartphone during transport. This guide teaches users about how to remove and re-insert the smartphone.

Appendix C: User Feedback

C.1. Electronic Controller Qualitative Feedback

ID	Type	Feedback	Heuristic	Related User IDs
Selection and Submission Controls				
1	+	The user felt that the controller was fast, efficient and accurate. The user felt in control.	User Control and Freedom	3, 4, 7, 8, 11, 12, 15, 19
2	+	The user felt that the point and click mechanism was easy to understand and simple to use.	Flexibility and Efficiency of Use	3, 12, 20, 17, 27
3	-	Felt that the ease of submission made their submissions more error prone as they did not have to concentrate as much with this controller (for participant 7, they scored the same in both attempts).	User Control and Freedom	7
Magnification Controls				
4	+	The magnification mechanic gave the user a sense of control.	User Control and Freedom	3, 7, 9, 26, 27
5	+	The user felt that using head tracking for the control of the magnification window was natural and immersive.	Match between System and the Real World	7, 20, 23
6	-	The user was frustrated by the magnification mechanic or found it challenging to use, especially when trying to find a flying bird or swimming fish.	User Control and Freedom	8, 12, 15, 17, 18, 27
7	-	The shape of the controller coupled with the users experience with desktop games (and inexperience with VR games) resulted in the user feeling disorientated initially as they expected that the controller's movement would control the magnification view rather than their head's gaze direction.	Match between System and the Real World	22, 26
8	-	The user felt that the zoom magnitude should be higher.	User Control and Freedom	8, 11

9	-	The user accidentally pressed the magnification button at the start of the game and did not realised that it was open until a few minutes into the evaluation. (Observation: This issue was difficult to detect in a scenario where the instructor could not see the view of the participant.)	Visibility Of System Status	21
Physical Controller Implementation				
10	+	The controller has a familiar design (e.g., like a smart TV controller)	Recognition over Recall	4, 11, 19
11	+	The controller was easy to hold and use.	Flexibility and Efficiency of Use	3, 17, 25
12	-	Controller was not positioned where it was expected to be. (Observation: Non-positional tracking was initially perceived as an error).	Immersion	2, 8, 14, 17, 19
13	-	The user felt that it was difficult to remember the controls. Specifically, they swapped the selection and magnification clicks.	Recognition over Recall	5, 7, 20, 27
14	-	Controller did not connect when the user started the game. (Note: This was an issue with reliability, the controller would be powered on while users were watching the tutorial videos and in a few instances it would not be connected to the phone by the time the user started playing, a full Bluetooth pairing process would then need to be carried out.)	Immersion	1, 6, 19
Interface Design				
15	+	The way in which questions and possible answers were displayed was easy to follow.	Consistency and Standards	12
16	-	The zoom mechanic lacks a frame of reference which makes it difficult to aim the view.	Error Preventions	19
17	-	The auditory feedback and visual highlighting was not enough to signal that a question was answered.	Help Users Recognize, Diagnose and Recover From Errors	27

Interface Accessibility				
18	+	The pointer beam assisted the user in reading the questions on the question board.	Flexibility and Efficiency of Use	7
19	+	Images used on the question board were of sufficient quality, this was useful given that the board was on the other side of the room.	Aesthetic and Minimalist Design	16
20	-	Felt less eye strain with this magnification approach as there is no need to focus on the centre of the screen.	Immersion	20
21	-	It was more difficult to read names in this approach.	Aesthetic and Minimalist Design	19
Controller On-Boarding				
22	+	The user made use of the rewind feature while watching the interface's tutorial video to get further clarity on its usage.	Help and Documentation	3, 16
Overall Experience				
23	+	"I was unaware of my surroundings and immersed in the game."	Immersion	3, 5, 6, 9, 13, 17, 21, 22, 27
24	+	The user did not feel time pressured.	Immersion	3, 7, 8
25	-	User felt that the efficiency of the controller took away from the immersion. Reasons for this includes boredom, and too much focus on answering.	Immersion	15, 19, 27
26	-	This interface encourages visual matching over learning the names of the plants and animals.	Immersion	4, 19

C.2. Computer Vision Controller Qualitative Feedback

ID	Type	Feedback	Heuristic	Related User IDs
Selection and Submission Controls				
1	+	The selection and submission controls were easy to understand and added to the immersion.	Match between System and the Real World	7, 8, 9, 27
2	+	Appreciated that the selection wraps around when scrolling past the end of the selection options.	Match between System and the Real World	6, 7, 22
3	-	The placement of the submission box made it more challenging to use. It was considered to be too far away. This was especially true if the participant was testing on a couch.	Flexibility and Ease of Use	2, 10, 12, 13, 14, 16, 17, 18
4	-	User tried touching the submission box with the tip of the controller rather than the whole control (i.e. keeping the controller facing the camera) which reduced tracking fidelity.	Flexibility and Ease of Use	3, 18
5	-	The submission mechanic felt cumbersome.	Flexibility and Ease of Use	2, 19
6	-	The user forgot how the submission mechanic worked at the start of the game.	Recognition over recall	19, 24
7	-	The user scrolled too quickly causing the selection to skip an option.	Consistency and Standards	6
Magnification Controls				
8	+	The magnification mechanic (i.e. flipping the controller) was easy to understand and added to the overall immersion.	Flexibility and Ease of Use	4, 8, 9, 11, 12, 13, 14, 16, 17, 18, 19, 20, 24, 26, 27
9	+	The user preferred the vision controller's magnification mechanic over the electronic controller one as it was more intuitive.	Match between System and the Real World	11, 14, 19, 27
10	-	It was difficult to know where the magnified view was directed. This made aiming it more difficult, especially when trying to see a flying bird.	Error Preventions	8, 22, 25, 26
11	-	User initially forgot about the magnification mechanism.	Recognition over recall	1, 24

12	-	Zooming in hides the question and possible answers which places a greater memory load on users.	Recognition over recall	17
Physical Controller Implementation				
13	+	Controller was easy to hold and felt familiar in the hand.	Match between System and the Real World	3, 27
14	-	The amount of turning the gear requires before a change in selection is registered is more than participants expect. This was particularly frustrating when trying to scroll to the last option (and wrap around was not used).	Flexibility and Ease of Use	4, 7, 8, 9, 11, 12, 13, 14, 16, 17, 18, 20, 25, 26, 27
15	-	The gear selection not being actively tracked makes the controller feel unresponsive. It is unclear if tracking is working or not.	Error Preventions	2, 9, 11, 12
16	-	The controller tracking felt jittery when moving the head or when the controller was not in the centre of focus.	Match between System and the Real World	6, 8, 17, 19
17	-	Felt that this approach requires more movement than expected from this type of exercise.	Aesthetic and Minimalist Design	10
Interface Design				
18	+	The submission box animations and sounds added to the immersion.	Visibility of System Status	7
19	+	Appreciated that the selection was kept when switching to the magnifier view and back.	Match between System and the Real World	27
20	+	Having the controller disappear was helpful for being able to inspect what is behind it.	User Control and Freedom	27
21	-	Did not see submission confirmation as the controller lost tracking just after submission causing the submission animation to cancel.	Consistency and Standards	5, 22
22	-	The way the controller disappears when tracking is lost is jarring for the user.	Consistency and Standards	2

Interface Accessibility				
23	+	The (non-blurred) view surrounding the magnification tool provides a frame of reference that allows users to correct their aim when using the magnification mechanic.	Error Preventions	19
24	+	Being able to bring the controller closer made the reading of questions easier.	Error Preventions	8
25	+	Images used on the question board were of sufficient quality, this made understanding the question easier.	Aesthetic and Minimalist Design	16
26	-	The submission boxes were initially not noticed because the user had poor peripheral vision.	Visibility of System Status	4
Controller On-Boarding				
27	+	The user made use of the rewind feature while watching the interface's tutorial video to get further clarity on its usage.	Help and Documentation	14
28	-	The user was not sure how to rotate the selection gear initially as they had a VR headset on before they were given the physical controller. The video tutorials were not enough to convey this information.	Help and Documentation	3, 16, 23
29	-	The user did not initially understand that having the controller in the line of sight of the phone's camera was important for tracking.	Help and Documentation	1
30	-	The user needed reminding that the question was displayed on the virtual representation of the controller even though the introduction video mentioned this.	Help and Documentation	21
Overall Experience				
31	+	"I was immersed in the game and unaware of my surroundings."	Immersion	1, 7, 16, 19, 21, 22, 27
32	+	The user felt that the controller was more fun and immersive than the electronic controller.	User Control and Freedom	4, 10, 15, 19
33	-	The user felt that there was a bigger learning curve for this controller and that it was more difficult to use in general.	Immersion	7, 10, 13, 21, 22

C.3. Educational Game Qualitative Feedback

ID	Type	Comment	Heuristic	Related User IDs
Game Design: On-Boarding				
1	+	User made use of the rewind feature in the tutorial video since it was played using a video player of the evaluation smartphone.	User Control and Freedom	3, 25
2	-	It was difficult to distinguish between the different areas of the map at the start of the game. (Observation: The concept of different areas was initially unclear, even though it was explained in the tutorial videos, the in-game voice prompt and the in-game area key)	Recognition over recall	1, 2, 4, 5, 6, 8, 10, 11, 16, 17, 20, 21, 27
3	-	Did not understand that looking (i.e. aligning eyes) at the user ID was different from “gazing” (i.e. aligning head) at it.	Match between System and the Real World	5, 15
4	-	User felt overloaded with all the information presented to them during the video on-boarding as well as at the start of the game.	Help and Documentation	3, 9
5	-	The user did not realise when the tutorial ended and the main evaluation started.	Visibility of System Status	14
Game Design: Gameplay				
6	+	It was an enjoyable and relaxing experience.	Immersion	1, 2, 6, 9, 10, 13, 14, 17, 21, 25
7	+	The viewpoint was enjoyable and made the user feel like a deity watching from above.	Immersion	3
Game Design: Educational Implementation				
8	+	The level of difficulty was correct and rewarding.	Consistency and Standards	3, 7, 8, 16, 21
9	+	The user felt that the timer added positively to the challenge of the game.	Consistency and Standards	16
10	+	User appreciated learning the names of plants and animals.	Immersion	11
11	-	Felt that the time pressure created by the timer took away from the relaxed atmosphere created by the game.	Consistency and Standards	3, 21

12	-	The questions about what plants grew <i>adjacent</i> to the river were not obvious in meaning.	Help and Documentation	5
13	-	The user wanted the questions to be more challenging.	Consistency and Standards	13
14	-	User desired more accompanying information for plants and animals.	Help and Documentation	3
15	-	The user did not notice the timer at all.	Visibility of System Status	11
Game Design: Aesthetics				
16	+	The graphics were aesthetically pleasing and helped to encouraged exploration.	Immersion	3, 4, 7, 8, 12, 13, 17, 23
17	+	The variety and animations of the animals added to the overall immersion.	Immersion	4, 7, 13, 14, 18, 20
18	+	The music and sound effects were pleasant and added to the immersion.	Immersion	7, 8, 14, 19
19	+	Animations between map changes were pleasant.	Immersion	3, 18
20	+	The VR room's décor and the views from its windows added to the atmosphere.	Immersion	3, 7
21	+	The user found the area key useful.	Help and Documentation	5, 13
22	-	It wasn't clear where the sky was.	Consistency and Standards	18, 20
23	-	The colour choices for the fish and the forest biome made for more ambiguity.	Consistency and Standards	20, 27
VR Headset Usage				
24	+	The experience was novel and pleasing.	Immersion	4, 11, 20, 25
25	+	The headset was considered light in weight and thus did not impact long term use.	Immersion	11
26	-	The FOV felt too close.	Match between System and the Real World	4, 5, 19
27	-	User felt eye strain related to the smartphone's display's resolution.	Immersion	11, 20, 23
28	-	The use of a face mask (due to Covid-19 protocol) while using the VR headset resulted in the lenses of the headset becoming misted. This was corrected by adjusting the mask's position.	Immersion	3, 17

29	-	The headset was initially uncomfortable to wear.	Immersion	13
30	-	A user who wore glasses during the gameplay found that their eyes would go out of focus occasionally.	Immersion	10
31	-	User could not see timer due to poor peripheral vision in the left eye.	Immersion	4
32	-	The in-game avatar (i.e. the grey capsule shape) models a standing player while most players test while sitting. This creates an inconsistency with where the user feels their legs are placed.	Consistency and Standards	14