Department of Computer Science & Informatics
(CSIS6809)

Technical Manual

Virtual Reality Computer Assembly:

Building Technical Skills Through Immersive Learning

Name: N.E. Barkhuizen

Student No.: 2019602537

Project No.: 2023/14

Supervisor: Prof. L. de Wet

Co-supervisor: Dr. B.S. Botha

Virtual Reality Computer Assembly:

Building Technical Skills Through Immersive Learning

Project Number: 2023/14

CSIS6809 Honours Project

Technical Manual

Submitted as partial requirement for the degree of BSc (Honours)

Department of Computer Science & Informatics

Faculty of Natural and Agricultural Sciences

University of the Free State

Nadean Elizabeth Barkhuizen

2019602537

Project Advisor: Prof. L. de Wet

Co-Advisor: Dr. B. S. Botha

November 2023

Table of Contents

List of Figures

List of Tables

Abstract

This research project aims to develop a virtual reality (VR) application to facilitate computer hardware education for students. With the help of the VR application, students will be able to identify computer components, understand their functions, and learn how to assemble them into a functional computer. The proposed application will incorporate interactive elements that allow students to explore and investigate computer components virtually and receive real-time performance feedback. The project will focus on evaluating the effectiveness of the VR application as an educational teaching tool, as well as the user experience. The VR application offers an interactive approach to learning about computer components and assembly, which has the potential to enhance the educational learning experience.

Chapter 1: Introduction

1.1 Introduction

The use of virtual reality (VR) technology in education has been gaining traction in recent years due to its potential to create immersive and engaging learning experiences (Leung, Zulkernine, & Isah, The use of Virtual Reality in Enhancing Interdisciplinary Research and Education, 2018). With the increasing demand for digital literacy and computer education, there is a need for innovative and effective ways to teach students computer skills like computer hardware and assembly. This research proposal aims to develop a VR application that will allow students to learn and practice computer components and the assembly of a functional computer.

The proposed VR application will enable students to explore the small world of computer components and assembly by providing an interactive and immersive learning environment. The application will feature realistic three-dimensional (3D) models of computer components and a virtual workspace where students can assemble the components. Additionally, the application will provide educational resources such as instructional materials to assist students during the learning process.

This research also aims to evaluate the VR application's effectiveness in teaching students' computer hardware and assembly skills, as well as their user experience. The evaluation will include measuring student learning outcomes and engagement levels, as well as gathering feedback on the usability of the application.

The development of this software has the potential to fundamentally alter the way computer literacy skills are taught in educational facilities. Students can develop a deeper understanding and appreciation for computer hardware and assembly through hands-on, interactive, and immersive learning experiences, which can ultimately improve their digital literacy and computer skills.

1.2 Problem Statement

The main problem revolving around computer assembly is that students may lack the confidence needed to assemble a functional computer. Overall, the assembly of a functional computer can become complex, since it requires precision and caution.

Most computer components consist of small pins which can be damaged quite easily, even with extensive knowledge and experience in computer assembly. This puts unnecessary pressure on students as there is no room for error. Providing students with a virtual environment will enable them to learn and make mistakes without real-world repercussions.

The question of accessibility should also be considered in today's environment. It is realistic to assume that physical computer hardware components are expensive and difficult to obtain. Therefore, offering each student the opportunity to assemble a fully functional computer in a lab seems unattainable compared to training them via a VR application.

Computer assembly is also an abstract concept. It can be difficult to understand the compatibility and function of most components without visualising how they interact in a functional environment.

Addressing these challenges through interactive and immersive teaching methods, providing practical experience, and access to resources can help students overcome these challenges and become successful in learning about computer components and assembly.

1.3 Scope of the project

The proposed application will be developed to address the problem of ineffective teaching methods, specifically in the field of computer hardware. This application will encapsulate a specific set of features and functionalities.

Firstly, the VR application will provide a virtual environment that replicates a real-world computer assembly scenario. This environment will include features like a virtual workspace with the necessary components that are needed for a typical computer assembly scenario.

This application will provide interactive tutorials to guide students through the process of identifying computer components, learning about their functions, and how to assemble them into a working computer. These tutorials will be engaging and easy to follow for beginners in the field of computer assembly.

It is extremely important that the developed computer components and environment are realistic, three-dimensional replicas. These models will be accurate and visually appealing, allowing students to identify them easily and correctly within the application.

Lastly, this application will contain some form of helpful tool that will provide students with tips and any additional necessary information that will lead the students to a successfully assembled, functional computer.

1.4 Technologies involved

Based on research, the most popular game engine that supports VR development is Unity 3D. This engine has an intuitive and easy-to-learn language code with a large community of game developers who prefer this platform. It also utilises C# as a programming language, which is popular when it comes to software development. Unity offers a wide range of assets in their store, which is another reason why Unity is highly recommended by developers (Hussain, Shakeel, Hussain, Uddin, & Ghouri, 2020).

In terms of hardware, the Oculus Quest (Figure 1) is a suitable VR headset that offers an immersive and wireless VR experience. The Oculus Quest is a standalone VR headset that does not require a PC or any additional hardware to operate, making it a convenient and portable option for VR users. It has built-in positional tracking, allowing users to move around in the virtual environment without any external sensors. The use of the Oculus Quest is highly advisable as it will be difficult to work with and assemble computer components virtually while using hand controllers (LaRocco, 2020).

Figure 1: Oculus Quest (Amazon, 2023)

1.5 Required skills and challenges

The researcher should be proficient in using Unity as VR development software to create a 3D environment that accurately represents a workspace and computer components. The logic of the application will be created by using the C# programming language; therefore, knowledge of this language is indispensable.

The functionality of the Oculus Quest headset, and its interaction via Unity, is also of paramount importance. As the researcher has no prior experience with VR hardware, this is a skill that needs to be acquired.

Furthermore, knowledge and understanding of computer hardware components and how they function, is essential to create an accurate VR simulation.

It is crucial that the researcher has knowledge of successful teaching methods. The software will lose its purpose if the method of teaching is vain and ineffective.

Lastly, some knowledge of user experience design, testing and debugging, and 3D modelling will become apparent. If the repository that consists of 3D models of computer components are incomplete, the researcher would need to create these models in a 3D creation software such as Blender.

1.6 Environmental impact

The main concern for a VR application that will facilitate a great number of students, will be the occurrence called Cybersickness, or virtual reality sickness. It is a type of motion sickness that transpires when using virtual reality devices. It is believed to be related to the mismatch between the sensory inputs that the human brain receives while in a virtual reality environment, and what the body experiences in reality (Nurnberger, Klinger, Witte, & Brodoehl, 2021).

This phenomenon is of concern as it can affect 20% - 95% of users (Caserman, Garcia-Agundez, & Gamez Zerban, 2021). This can discourage users from using the application and essentially eliminate the potential of the software to educate students on computer components and assembly.

It should also be noted that some users might be apprehensive towards the headset that will be used for the application, since they are unfamiliar with the technology. The headset could also be uncomfortable for some users, which could add to this issue.

On the other hand, the development and implementation of this application could indeed have a positive impact. This system could impact the students as well as the Department of Computer Science and Informatics in the following ways:

- Enhanced learning experience: This application could provide the students with a more engaging and immersive learning environment that they can interact with.
- Reduced costs: Even though the VR equipment is considered expensive, these purchases only need to be made once. Afterwards, the equipment can be used by multiple students throughout the year. This could be considered cost effective compared to purchasing multiple computer components and replacing them if damage is inflicted by inexperienced students.
- Improved safety: The process of assembling a computer can be dangerous. Not only are the students put at risk, but costly computer parts are also at risk of being damaged, especially if the students are inexperienced and not supervised properly. This system will omit this risk and provide a safe environment where students can practice regularly with no consequences of damaging expensive computer parts. It is still, however, important to supervise students while using the VR equipment, as it is also fragile equipment.

1.7 Methodology

The Agile Software Development methodology is a methodology that foresees the need for flexibility. It incorporates a level of pragmatism into the product deliverables, meaning that the focus is placed on the practical value of this methodology. This procedure involves breaking down projects into smaller, more manageable tasks and completing them in short iterations or sprints. During each sprint, the main objective is to design, develop, test, and deliver a working software increment that meets the client’s needs (Brush & Silverthorne, 2022).

In adapting Agile for one person, the researcher will need to keep track of process flaws to address at the end of the sprint. While the researcher will not be improving the collaboration with other team members, they can still improve on their own personal process. This will be made easier by keeping record and documenting throughout the sprints.

An Agile methodology approach for individuals, called Personal Scrum, adapts and applies scrum principles to one-person projects. It promotes personal productivity through observation, adaptation, gradual elaboration, time boxing, and prioritising and sizing work (Pahuja, 2015). This type of methodology is capable of handling the iterative and rapidly evolving nature of VR software development for individuals in the field.

1.8 Timeline

The expected timeline for each phase of development is broken down into more manageable parts below.

Requirements & Documentation:

Roughly, the requirement and documentation will be partially completed from April to mid-June as shown in figure 2. The documentation that will be completed includes chapters one through four of the Technical Manual.

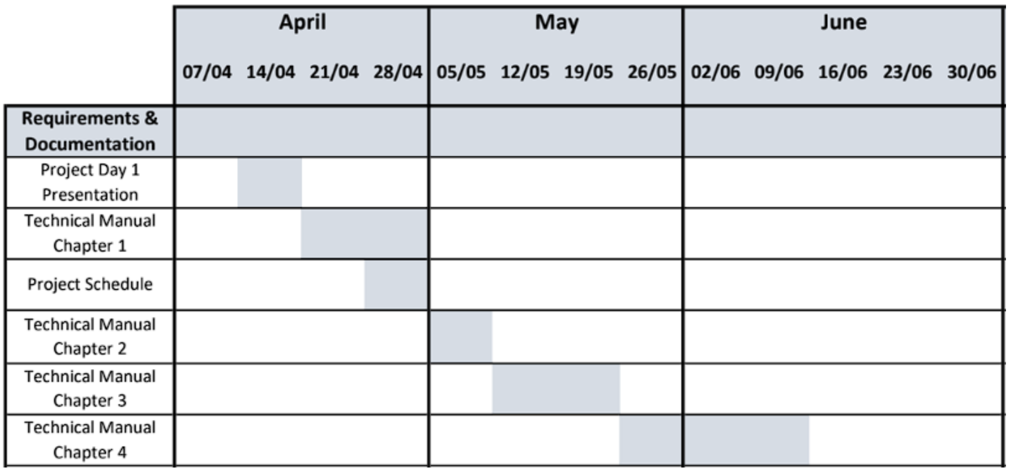


Figure 2: Requirements & Documentation Timeline Part 1

The requirements and documentation workflow will be revisited near the end of August to complete the final chapters of the Technical Manual as well as complete the three chapters of the User Manual. These will be reviewed and refined late-October before the submission date (Figure 3).

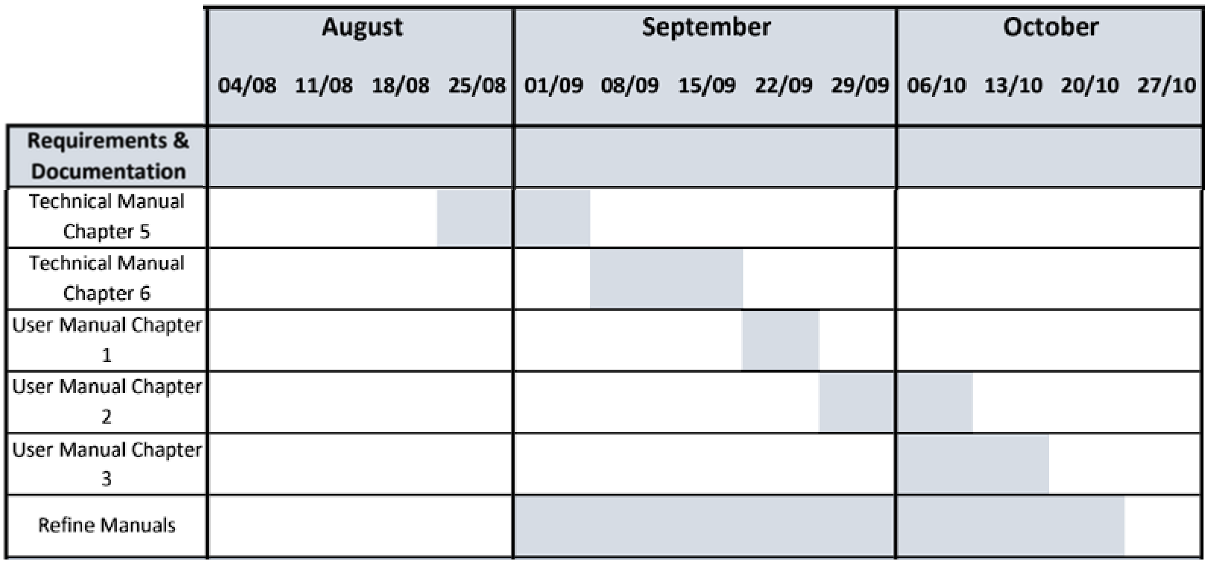


Figure 3: Requirements & Documentation Timeline Part 2

Design:

Some of the Design workflow will commence in mid-May and will continue until mid to late-June. The workspace, component models as well as the missing component models will be designed during this time as shown in figure 4.

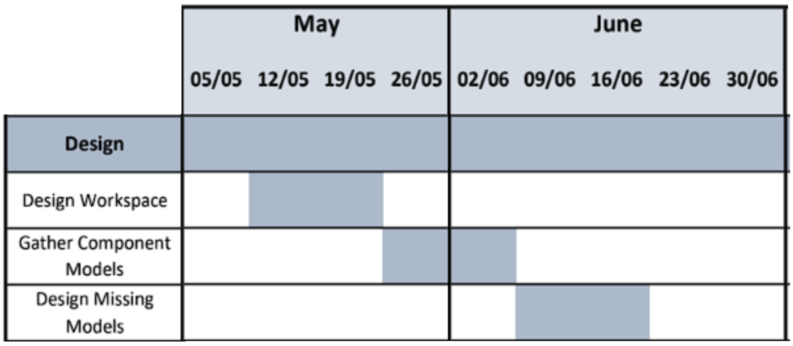


Figure 4: Design Timeline

Development:

The development phase, which is the longest, continuous phase, will stretch from late-June until the end of August (Figure 5). In this phase the workspace and the models will be implemented, and the scripts will be written for these objects.

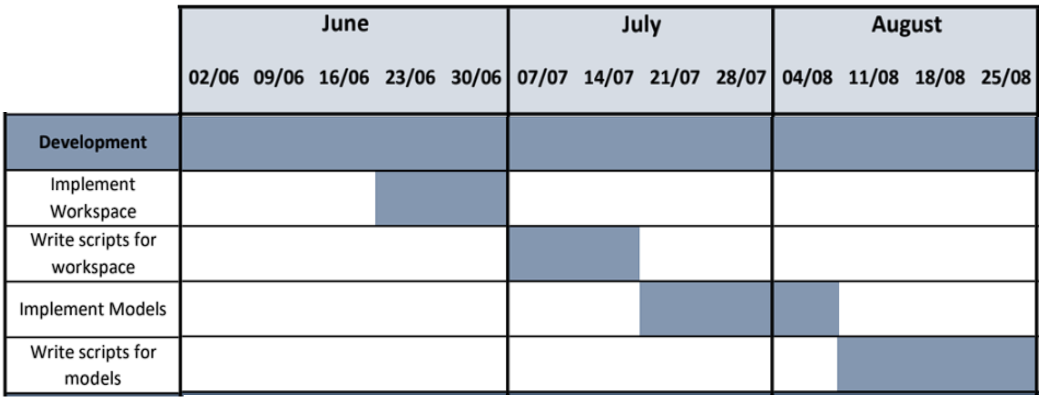


Figure 5: Development Timeline

Testing:

The testing workflow will allow the time to do a usability test followed by a number of weeks where improvements to the system can be made before the submission date. This workflow will run for the whole month of September as shown in figure 6.

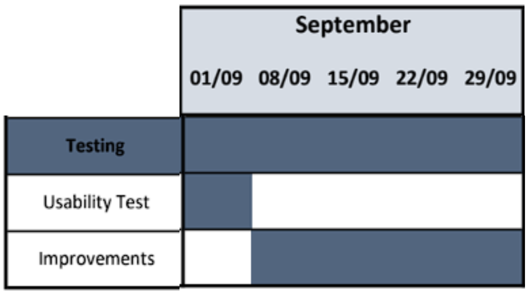


Figure 6: Testing Timeline

1.9 Limitations of the project

Due to the nature of developing and running a VR application, there are a few important hardware components to consider.

For the VR environment to render in real-time it will require a strong graphics card. A dedicated GPU with at least 8GB of RAM is recommended for the development of a VR application.

To increase the operability of the software, a CPU with a high clock speed is recommended to quickly model the environment. Additionally, the more cores the CPU possesses, the better, since the workload can be dispersed across multiple cores.

Developing a VR application consumes a lot of RAM, especially in this scenario where multiple applications will be used at the same time during development. Therefore, the recommended amount of RAM needed for this type of project is at least 16GB of RAM.

The unique feature of VR is all movements are occurring in real time. The software image must move identically to the user's movements; therefore, the environment must immediately respond to instructions. This software must be reliable and robust for it to effectively educate students on computer hardware and assembly.

Furthermore, it should be modelled against a real-world scenario to ensure that it is intuitive to use. It should be noted that computer components can be small and difficult to assemble, especially when users are new to a virtual environment. Therefore, an adaptation must be made that will alter the real-world feel of the environment. It might be necessary to enlarge the virtual environment beyond realistic measures to omit the possibility of users becoming frustrated with the equipment and the computer component models.

Lastly, due to time constraints, the focus of the project is placed on a guide for assembling computers coupled with information on the components, rather than a step-by-step focused application.

1.10 Document overview

Chapter 1 introduced the project's scope, technologies, challenges, and methodology to be used. Chapter 2 delves into a thorough literature review, examining VR history, applications of VR, and existing systems, leading to the proposal of the educational system. Chapters 3 and 4 dissect project requirements, analysis, and technology choices. Chapters 5 and 6 details the design aspects, emphasising spatial layout, environment interaction, and component integration.

Chapter 7 dives into evaluation strategies, metric analysis, and user feedback. Chapter 8 includes the source code and its implementation within Unity. Chapter 9 candidly discusses challenges faced, including Unity's learning curve and technical obstacles. Chapter 10 provides a concise conclusion, summarising the project's transformative journey.

Chapter 2: Literature Review

2.1 Introduction

Virtual Reality (VR) has rapidly emerged as a cutting-edge tool for training and education. Due to the realistic and interactive environment that it creates, it has distinct advantages over conventional teaching techniques (Leung, Zulkernine, & Isah, The use of Virtual Reality in Enhancing Interdisciplinary Research and Education, 2018).

In this literature review, the focus is placed on the use of VR technology in computer hardware education. The existing literature in this field is explored, including its application, effectiveness, and limitations.

2.2 Virtual Reality (VR)

2.2.1 History of VR

The history of VR technology dates to the 19th century. However, it did not gain popularity until the 1980s and 1990s with the development of more powerful computer graphics and processing capabilities (Cipresso P. , Giglioli, Raya, & Riva, 2018). In 1985, Jaron Lanier introduced the concept of “Virtual Reality” and established VPL Research, which developed and marketed VR products (Ambrosio & Fidalgo, 2020). Commercial VR products, such as the virtual arcade machines and the Nintendo Virtual Boy system, were introduced in the early 1990s. However, the accessibility and appeal of these early VR systems were limited by their high cost and weight (Stengel & Furman, 1996).

A resurgence in interest in VR occurred in the late 1990s and early 2000s as a result of the creation of more accessible and powerful computer hardware and software. Companies like Sony and Oculus started developing VR headsets that were more compact, comfortable, and capable of providing an immersive experience. Today, VR technology is growing rapidly, with a wide range of uses in industries, like entertainment, education, healthcare, and training. The creation of standalone VR headsets, which can be used without a computer or gaming console, has received greater attention, opening VR to a wider range of users (Hamad & Jia, 2022).

In general, there have been periods of invention and progression, followed by smaller periods of development that characterised the history of VR. Despite these obstacles, the technology has advanced, and VR's uses show potential, making it a fascinating and promising field for the future.

2.2.2 VR Terminology

VR has its own set of terminology that is unique to the field, as well as some that are shared in the field of Computer Science. The following terms will frequently be encountered during this technical manual. The terminology is defined below in an elementary way.

- VR (Virtual Reality) – the computer-generated simulation of a three-dimensional image or environment that can be interacted with in a seemingly real or physical way by a person using special electronic equipment, such as a Head-Mounted Display (HMD) or controllers (Lowood, 2023).
- HMD – a device worn on the head that displays virtual content to the user.
- Oculus Quest – an HMD device used for the development of this application.
- VR Input – the devices and methods used to input data into a virtual environment, such as controllers and hand tracking.
- VR Output – the devices and methods used to output data from a virtual environment to the user, such as HMDs.
- Unity – a popular game engine that is commonly used to develop VR applications and experiences.
- Immersion/Immersive – the feeling of being fully surrounded by and involved in a virtual environment.
- Tracking – the process of following the movements of the user's head, hands, or other body parts to update the virtual environment in real-time.
- Hand Presence – the feeling of having one's hands represented accurately in a virtual environment.
- Cybersickness – a form of motion sickness causing a feeling of nausea, disorientation, or discomfort that can occur in some users during or after VR use (Nesbitt & Nalivaiko, 2018).

These are only a few examples of terminology that can be encountered in the field of VR. As VR continues to evolve and new technologies are developed, new terminology will likely emerge.

2.2.3 Applications of VR

VR technology has a wide range of applications across many different industries. It has the ability to provide immersive experiences that can transport users to new worlds and create new experiences. From gaming and entertainment, to healthcare and education, VR has the power to attract and captivate users like no other technology.

The following applications of VR can be observed across a variety of industries mentioned above:

- VR is most popular in the gaming and entertainment field. It is often used to create immersive gaming experiences that allow players to feel as if they are living inside the game world. It is also used by museums, theme parks, and other entertainment companies to create interactive and immersive experiences.
- Closely following the gaming and entertainment industry, VR is commonly used for training and education purposes due to its ability to omit real-world risks that accompany real-life situations, for example aviation training and even medical surgery training. It can also be used to create an immersive educational experience, such as a virtual tour or a historical recreation.
- VR is often used in the healthcare field for a variety of purposes, most commonly for pain management, physical therapy, and phobia treatments.
- A closely related field that also utilises VR, is the sports and fitness industry. VR creates an immersive training experience, as well as some entertainment through VR sports games and simulations.
- A few other lesser-known applications of VR could be found in the field of architecture and design, where it is used to create 3D models of buildings or structures. This allows users to walk through and experience their designs. In the field of tourism and travel, users can take a virtual tour of popular tourist destinations, allowing them to experience these places without leaving their homes.

These are just a few examples of the many possible applications of VR technology. As VR continues to progress and evolve, it will become more accessible. This will lead to an even more diverse application of VR technology in the future.

2.2.4 Educational Requirement

VR has been increasingly employed as a strong tool in the field of computer science education. It offers unique and engaging ways to teach various concepts and skills for example simulating coding scenarios, complex algorithm visualisation, cybersecurity training and data analysis (Pirker, Dengel, Holly, & Safikhani, 2020)

It is valuable to assume that any student who enrols for a course in computer science will be educated and assessed on computer hardware concepts. Therefore, it is important to understand the value that this system will bring forward in terms of education. The envisioned system aims to function as an aid in teaching computer hardware concepts, meaning that existing foundational knowledge will elevate the user experience.

Additionally, the nature of the system assumes a dual-functionality characteristic as an educational and assessment tool. It is also designed to adhere to industry standards regarding the preparation, tools, assembly, and safety protocols that need to be enforced during the learning process. This will ensure that the level of teaching received is valuable to the user and of quality.

2.3 Existing Systems

Since VR in education and training is not a new concept, there are a few existing systems in the field of Computer Science. Below are the most relevant existing VR computer assembly systems that this literature review discovered.

2.3.1 PC Building Simulator

PC Building Simulator is a popular simulation game developed to give users the opportunity to simulate building a computer. Users can learn about the various components of a computer and how they work together in this realistic and educational environment.

It was developed by UK-based indie developer, Claudiu Kiss, and was initially released in 2018 for PC-use with no VR abilities. It has since been expanded to other devices, such as consoles (The Irregular Corporation, 2018).

Functionalities

The users are presented with a virtual computer case with several components, including motherboards, CPUs, RAM and graphics cards. They are then tasked to assemble these components into a functional computer, paying close attention to cable management and efficient cooling. Users are enabled to experiment with different components and combinations to test how it affects the computer's performance.

Along with a free-build mode that enables users to experiment with various component combinations and construct their own PC, the game also features a tutorial mode that guides players step-by-step through the process of building a computer.

Users may also encounter a wide variety of common issues that occur while building a PC, such as component failure or incompatibility issues. These issues must be diagnosed and fixed to complete the build.

PC Building Simulator has gained popularity among gamers and computer enthusiasts as a fun and instructive approach to learning about computer hardware and assembly. Despite not being created with education in mind, the game offers a realistic and engaging experience that aids users in developing important technical skills.

Advantages

Since this system received positive reviews, it is imperative that the advantages of this system are further researched to fully understand its value in computer hardware education. The following advantages are noted (The Irregular Corporation, 2018):

- Flexible – this system allows users to assemble computers without any restrictions, which allows users to experiment with different components. This enables the system to be very flexible, and in turn, it adds value to its educational abilities.
- Low risk – the system provides an environment that is a simulation of computer assembly which mitigates any real-world risk. This is highly valuable, as computer components can be damaged extremely easily, especially by inexperienced users.
- Realistic – this system provides an extremely realistic simulation of the real-world process of building a computer. The ability to select your own

components, as well as the troubleshooting feature, helps to create the aspect of realism that is highly sought after.

- Interactive learning – this system provides an interactive learning experience that allows for experimentation and the effect thereof. This could lead to a deeper understanding and appreciation of computer assembly.
- Popular with good reviews - The game has received very positive reviews and was praised for its educational value, although it is not available in VR.

These advantages allow the system to be highly desirable by any kind of user that is interested in computer assembly, and it is important to consider these aspects in the development of the proposed system.

Limitations

Although this system has some great qualities, it has a couple of major shortcomings listed below (The Irregular Corporation, 2018):

- VR – this system is not available in VR. This cannot be regarded as a shortcoming in general, but compared to the proposed solution it will be considered as a limitation.
- Limited educational value – this system does not provide the necessary educational information on the components that the proposed solution will incorporate.
- Accessibility – this system requires users to have a computer with the necessary resources to run the system. It is also considered expensive to purchase on the popular platforms that it is made available on.

Overall, while this is a useful tool for learning and experimenting with computer assembly, it is lacking educational information on the components and most importantly is not available in VR.

2.3.2 PC Virtual Lab

PC Virtual Lab is a simulation game available on the popular gaming platform, Steam. It gives the users the opportunity to assemble computers on different levels to test their skills and knowledge in computer assembly.

It was developed by a team called 3DUBU from the University of Burgos in Spain. It was launched in mid-2022 and is available in VR. However, this application received mostly negative reviews on the gaming platform, which revealed that users are not satisfied with the application (JDTech Solutions, 2020).

Functionalities

Firstly, this system is available in VR and as a PC simulator. The system provides the user with an empty computer case in a workspace and a range of computer components typically used in computer assembly. The users can place these components on a “plate”, at which time it will display educational information about the component.

An avatar that provides instructions and acts as a guide during the assembly process, makes its appearance.

The developers claim that the system improved student satisfaction compared to traditional teaching methods. They also claim that it improved visual recognition, a deeper understanding of the material, and increased performance.

Advantages

Although the system received negative reviews, there are still aspects that grasped the users’ attention. These aspects should be considered and researched to understand its value in computer hardware education. The following advantages were noted (JDTech Solutions, 2020):

- Cost – this software is free on the popular gaming platform, Steam.
- Dual Platform – the main advantage of this system in terms of the proposed solution is that it is available in both standard and VR modes.
- Assessment – this system allows the user’s attempt to be graded in some way, and it provides feedback on the assembly attempt that the user made.
- Educational information – the system provides information on the computer components, which is highly valuable in terms of the educational aspect of the system.
- Low risk – this system allows users to learn about computer hardware and assembly without any real-world risk, such as damaging very expensive computer components.
- Avatar – the system has an avatar that guides the user during the assembly process and provides tips to ensure a successful assembly attempt.

- Realistic – the system provides a realistic experience of the computer assembly process.
- Interactive learning – this system provides an interactive learning experience.

These aspects provide an advantage over competitors in some aspects, which should be considered in the development of the proposed solution.

Limitations

The above system contains highly desirable advantages, but still received negative reviews. The following limitations most likely resulted in the users being unsatisfied with the system (JDTech Solutions, 2020):

- Controllers – this system utilises controllers to manipulate and work with the components. This leads to a lesser realistic experience and takes away from the immersive environment.
- Professionalism – this system was developed by a Spanish university team which is most likely the reason for grammar and spelling mistakes. This leads to unclear instructions and takes away from the learning experience.
- Realism – among other things, users complained that the components snap into place once it comes close to their intended position. This takes away from the realistic component that is vital in a VR application.
- Handling – users complained that the components are hard to pick up and manoeuvre, which makes handling very difficult. This adds to the lack of realism and professionalism.
- Flexibility – this system does not provide such a wide range of components to experiment with when compared to PC Virtual Lab.
- Troubleshooting – this system has no aspect of troubleshooting.

Although most of these complaints can be connected to the gaming community critiquing it in terms of a game and not an educational tool, it should still be considered when developing the proposed solution does.

2.3.3 PC Building Simulator 3D

PC Building Simulator 3D was developed by Kunhar Games in 2022. It is only available on the Google Play store and not on any of the other popular gaming platforms. It has over 100,000 downloads and received average reviews on the platform. This application (app) teaches people PC-building skills in combination with time management skills. The developers claim to have a realistic look and placement, meaning that the computer parts' positioning during the assembly process is close to a real-life situation (Kunhar Games, 2021).

Functionalities

The mobile app provides step-by-step instructions explaining the order in which parts should be assembled and provides useful information on identifying each part and its function.

The story leads as follows:

The user acts as a technician that receives orders from customers that require the user to build various computers. The user needs to accept these orders and start assembling the requested components. The system includes an operating system (OS) function where the user must install the chosen OS onto the successfully assembled computer. The user is also required to install the necessary applications and drivers onto the computer.

The app has in-game currency that is used to purchase components, and the user gets paid for the completed orders. Mini games are also available for play.

This app is not available on VR or on a PC.

Advantages

Although this is a mobile application, there are a few advantages to consider when developing the proposed solution (Kunhar Games, 2021):

- Educational information – this app provides step-by-step instructions for the assembly process, as well as useful information on the computer components.
- Flexible – the app provides a small range of different components to use during the assembly process, although not as extensive as PC Building Simulator.
- Low risk – this app mitigates the risk that comes with computer assembly.
- Accessible – anyone with a smartphone can download this app and learn about computer hardware and assembly.
- Cost – this app is free on the Google Play store.

Limitations

There is a wide range of limitations that follow this system. It should be carefully researched and considered when developing the proposed solution. The following limitations were observed (Kunhar Games, 2021):

- VR – it is not available in VR, nor is it available on a PC platform.
- Reliability – users complain that the app crashes frequently.
- Realism – even though the components' design is realistically modelled, it lacks realism, because the user must swipe on the screen to assemble components.
- Troubleshooting –no form of troubleshooting is available in this application.

2.4 Summary of Existing Systems

The above-mentioned systems all have their own unique functionalities that distinguish them from their competitors. It is important that the proposed application includes most, if not all, of the advantages of all the systems. The limitations and shortcomings of these systems should also be noted to avoid them reoccurring in the proposed application.

Below in Table 1 is a comparison of the existing systems to the proposed system. The overall purpose of the proposed system is to educate students about computer components and assembly.

2.5 Proposed System

The proposed system is a VR software application that will address the limitations and shortcomings of the above-mentioned existing systems. It will contain a very specific set of functionalities and features.

Firstly, the VR application will provide a virtual environment that replicates a real-world computer assembly scenario. This environment will include all necessary components and helpful material on how to assemble a computer during training sessions. Ultimately, this application will allow students to practice and experiment with computer components without the responsibility of working with real-world components.

Table 1: Existing Systems Comparison

Functionality	PC Building Simulator	PC Virtual Lab	PC Building Simulator 3D	Proposed System
VR application	✗	✔	✗	✔
Multiple components to choose from	✔	✗	✔	✔
No real-world risk	✔	✔	✔	✔
Realistic components and manoeuvrability	✗	✔	✗	✔
Educational learning material	✗	✔	✔	✔
Feedback & assessment	✗	✔	✗	✔
Troubleshooting	✔	✗	✗	✔

This application will provide an interactive tutorial to guide students through the process of identifying computer components, learning their functions, and assembling them into a working computer. These tutorials will be engaging and easy to follow by any student, no matter their level of existing knowledge.

It is vital that this application's computer components are realistic, three-dimensional replicas. These models will be accurate and visually appealing, allowing students to identify them easily and correctly in the application.

Furthermore, this application will contain assessment tools to test students' knowledge of computer components and their ability to assemble a functional computer. These assessments should be interactive and provide valuable feedback to students.

2.6 Summary

Overall, the literature review suggests that VR applications can be a very effective and engaging tool for education. However, extensive research is needed to address the limitations and shortcomings associated with these applications. This research will build a fundamental understanding and foundation for the successful development of the proposed application.

Chapter 3: Requirements and Analysis Workflow

3.1 Introduction

In this chapter, use cases will be utilised to capture the interactions and scenarios of the VR application. The aim is to define the behaviour and functionality of the system from the users' perspective by creating use case descriptions and scenarios. This structured approach will assist with the gathering and analysis of requirements which, in turn, forms a solid foundation for development and testing.

3.2 Use cases

3.2.1 VR Environment Navigation

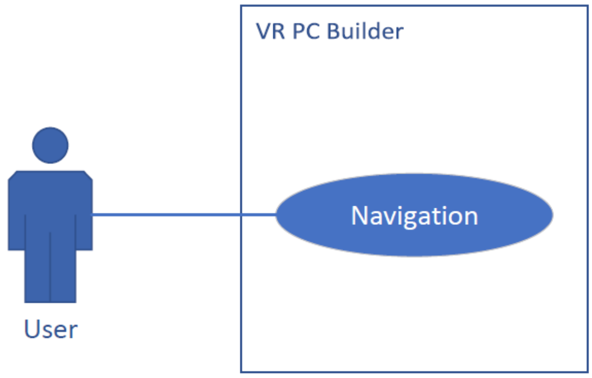


Figure 7: VR Environment Navigation Use Case

Table 2: VR Environment Navigation Use Case Description

Name of use case: VR Environment Navigation
Actors: User
Brief description: This use case covers the interactions related to navigating within the VR environment such as moving around and exploring the virtual space.
Step-by-step description: 1. The VR application is launched. 2. The user places the Oculus Quest headset on his/her head. 3. The user is immersed within the VR environment. 4. The user looks around the virtual environment. 5. The user moves around in the virtual environment.

3.2.2 Interacting with Objects

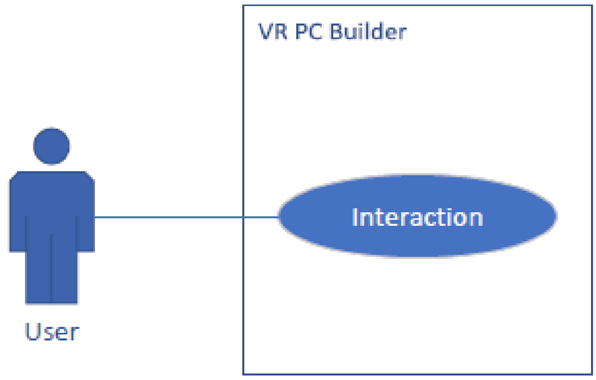


Figure 8: Interacting with Objects Use Case

Table 3: Interacting with Objects Use Case Description

Name of use case: Interacting with Objects
Actors: User
Brief description: This use case focuses on the interaction between the user and virtual objects, including the ability to pick up, manipulate or interact with the items in the VR environment.
Step-by-step description: Preconditions: • The VR application is launched and running. • The user is wearing the VR headset. • The user is immersed in the VR environment with interactive objects. 1. The user visually identifies the desired object in the VR environment. 2. The user physically reaches out with his/her hand/s toward the object. 3. The system detects the user's movement and/or interaction. 4. The system updates the user's movement and activates the object's response. 5. The user can manipulate the object based on its interactive capabilities. 6. The user performs the desired action on the object. 7. The system updates the object's position, orientation and/or state based on the user's manipulation. 8. If the object is released, it remains in the last known position. 9. The user can interact with any other 'interactable' objects throughout the environment, repeating steps 1-7.

3.2.3 Starting Scene

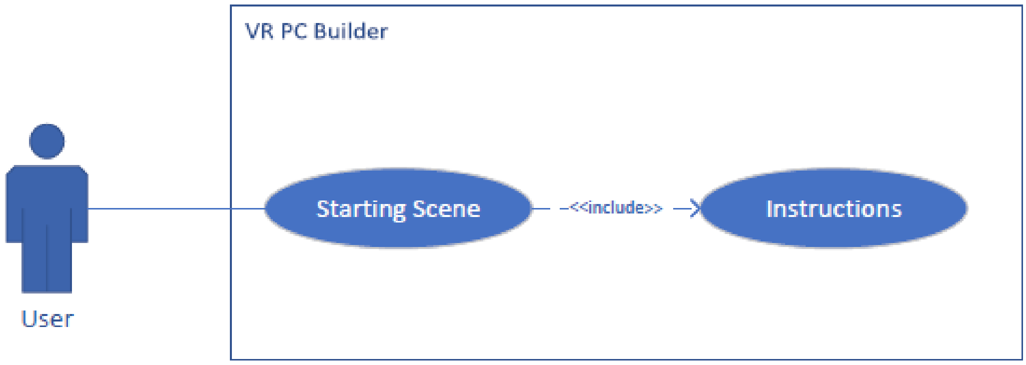


Figure 9: Starting Scene Use Case

Table 4: Starting Scene Use Case Description

Name of use case: Starting Scene
Actors: User
Brief description: This use case allows the user to find instructions on how to use the application including how to interact with the objects as well as the menu options to navigate the application.
Step-by-step description: Preconditions: • The VR application is launched and running. • The user is wearing the VR headset. • The user successfully entered the VR environment. 1. The system displays instructions within the VR environment via text and images to provide some guidance and orientation to the user. 2. The user reads the instructions provided by the system. 3. The user follows the instructions to understand controls, functionalities, and objectives within the VR environment. 4. The system allows the user to practice interacting with an object before continuing to the next scene. 5. Once the user understands the instructions and practised interacting with an object, he/she can proceed to the next scene in the VR application by selecting a button.

3.2.4 Practice Scene

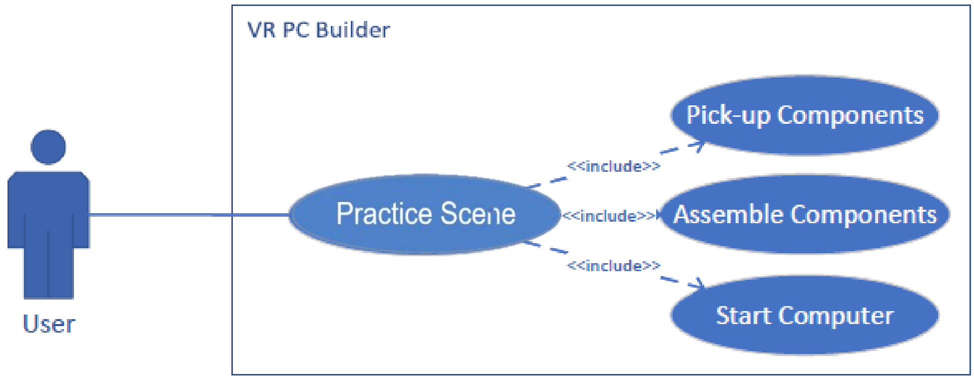


Figure 10: Practice Scene Use Case

Table 5: Practice Scene Use Case Description

Name of use case: Practice Scene
Actors: User
Brief description: This use case describes the process of a user assembling a functional computer within the VR application via interacting with the components provided.
Step-by-step description: Pre-conditions: • The application is launched and running. • The user is wearing the VR headset. • The user entered the VR environment. • The user successfully completed the starting scene. 1. The user is presented with a virtual workspace containing all the necessary computer components to assemble a functional computer with help material to guide them. 2. The user visually identifies a computer component from the available options. 3. The user interacts with the desired component by picking it up. 4. The user places the component correctly within the virtual environment. 5. The system validates this placement and reacts accordingly. 6. The user repeats steps 2-5 until all components have been assembled correctly. 7. As the user adds the components, the system keeps track of the assembly process and updates the environment accordingly.

8. The user connects the components via the necessary cables. 9. The system validates the cable connections if placed correctly and updates the environment accordingly. 10. The system validates the assembled computer after the last component has been positioned or the last cable has been connected. 11. The system responds by displaying a completion message with options to navigate the system further.

3.2.5 Assessment Scene

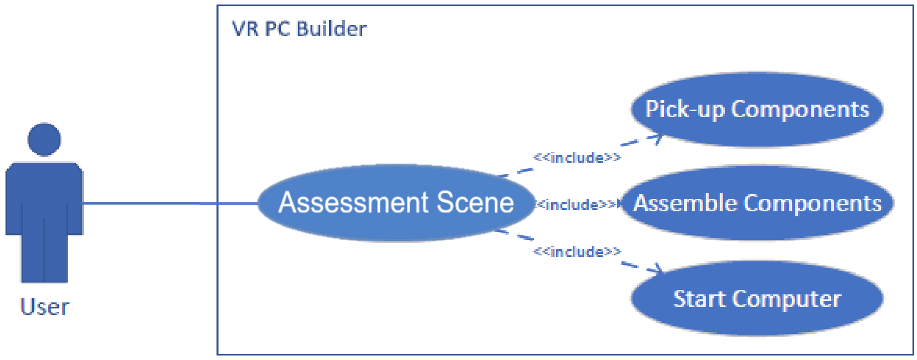


Figure 11: Assessment Use Case

Table 6: Assessment Scene Use Case Description

Name of use case: Assessment Scene
Actors: User
Brief description: This use case describes the results scene in the application where users can view the results of his/her computer assembly attempt and receive feedback on his/her performance.
Step-by-step description: Pre-conditions: • The application is launched and running. • The user is wearing the VR headset. • The user entered the VR environment. • The user successfully completed the starting scene and the Practice Scene.

1. The user is presented with a virtual workspace containing all the necessary computer components to assemble a functional computer with no help material to guide their assessment. 2. The user visually identifies a computer component from the available options. 3. The user interacts with the desired component by picking it up. 4. The user places the component correctly within the virtual environment. 5. The system validates this placement and reacts accordingly. 6. The user repeats steps 2-5 until all components have been correctly assembled. 7. As the user adds and removes components, the system keeps track of the assembly process and updates the environment accordingly. 8. The user presses the "Grade" button to assess their attempt. 9. The system displays customised feedback on their attempt with buttons to navigate the system further. 10. The user views the feedback and presses the desired button to navigate the system.

3.2.6 Use Case Diagram

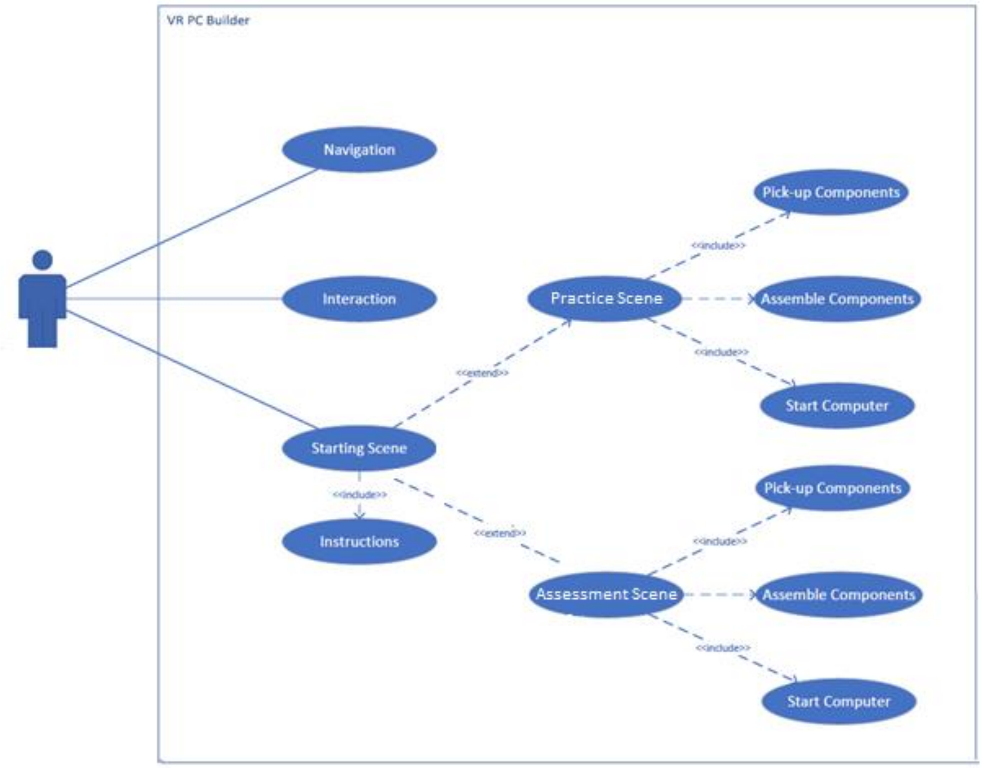


Figure 12: Use Case Diagram

3.3 Scenarios

3.2.1 VR Environment Navigation

Best case scenario:

1. Levi enters the VR environment and is quickly familiarised with the navigation controls and options.
2. Levi effortlessly moves around the virtual space.
3. The VR system accurately tracks Levi's movements and provides smooth and responsive navigation.
4. Levi easily explores different areas of the VR environment and seamlessly transitions between scenes.

Worst case scenario:

1. Eren enters the VR environment but struggles to understand the navigation controls.
2. Eren experiences difficulties in moving around the virtual space, encountering issues with tracking accuracy or delayed responses.
3. The VR system's navigation is complex and counterintuitive leading to confusion and frustration for Eren.
4. The overall navigation experience in the VR environment is frustrating, cumbersome, and hinders Eren's interaction and learning experience.

Alternative scenario:

1. Mikasa enters the VR environment but feels uncomfortable as this is a new experience for her.
2. After spending a bit of time in the VR environment, Mikasa becomes more comfortable with the hardware and the experience.
3. Mikasa explores the area more confidently and is satisfied with the navigational experience.
4. The system tracks Mikasa's movements and provides an average experience in terms of performance.

3.2.2 Interacting with Objects

Best case scenario:

1. Dominic enters the VR environment and is presented with a range of interactive objects.
2. Dominic effortlessly identifies and interacts with the objects using intuitive gestures.
3. The VR system accurately detects and responds to Dominic's interactions in real time, providing seamless and natural object manipulation.

4. The interactive objects offer realistic physics and haptic feedback, enhancing Dominic’s sense of presence and immersion.
5. Dominic manages to pick up, move, rotate, and place objects seamlessly.

Worst case scenario:

1. Craig encounters difficulties in interacting with the objects in the VR environment.
2. The VR system struggles to accurately detect and respond to the user’s interactions, leading to a delay in feedback.
3. The interaction controls are counterintuitive to Craig which leads to his confusion and frustration.
4. Craig struggles to pick up, move, rotate, and place objects which leads to an unsuccessful attempt at assembling a computer due to these issues.

Alternative scenario:

1. Elias struggles to understand the interactive controls at first.
2. This leads to Elias not being able to pick up, move, rotate, or place objects at first.
3. Elias takes his time to become proficient with interacting with the objects before moving on to the starting scene.
4. The system is moderately accurate at detecting Elias’ movements and updating the environment accordingly.
5. This leads to an average experience in terms of a realistic experience.

3.2.3 Starting Scene

Best case scenario:

1. Vicente is immersed in the starting scene that provides clear and concise instructions on how to navigate the environment and interact with objects.
2. The instructions are easy to understand and presented in clear, readable text, coupled with images to increase comprehension.
3. Vicente quickly grasps the controls.
4. Vicente interacts with the practice object seamlessly and becomes comfortable with the VR environment.
5. Vicente continues to the next scene by selecting the desired button, feeling confident in his abilities.

Worst case scenario:

1. Jalal is immersed in the starting scene that provides some instructions on navigation and interaction of the application.
2. Jalal struggles to understand the instructions.
3. Jalal tries to interact with the practise object but struggles to get comfortable with the controls.

4. Jalal does not feel comfortable or confident enough to enter the Practice Scene.
5. This leads to Jalal not completing the objective of the Practice Scene that follows.

Alternative scenario:

1. Zofia is immersed in the starting scene that provides adequately clear instructions on navigation and interaction of the application.
2. At first, Zofia does not grasp the controls.
3. Zofia tries to interact with the practise object but struggles to get comfortable with the controls.
4. After spending some time with the object, Zofia feels more comfortable but still a bit hesitant.
5. Zofia continues to the Practice Scene by selecting the button even though she does not feel fully confident in her abilities.

3.2.4 Practice Scene

Best case scenario:

1. Miles enters the Practice Scene and is presented a well-designed workspace to assemble the computer on.
2. Miles is provided with clear and concise instructions and tips along the way which guides him throughout the process.
3. The system accurately tracks Miles' movements and interactions, giving him a smooth experience.
4. Miles successfully assembles the computer within the building scene which increases his confidence.
5. Miles gains valuable knowledge and understanding of computer hardware through this process.

Worst case scenario:

1. Alexsandr enters the Practice Scene but encounters an unfamiliar workspace, making it challenging for him to identify and interact with the computer components.
2. The provided instructions and tips do not provide Alexsandr with the necessary guidance for him to confidently attempt the assembly process.
3. This leaves Alexsandr unsure and confused about what he needs to do.
4. Alexsandr struggles to visually identify the provided components leading to further confusion.
5. The system does not accurately track Alexsandr's movements, resulting in frustration and him further doubting his abilities.
6. Alexsandr struggles to complete the objective within the Practice Scene, resulting in task failure.

Alternative scenario:

1. Saif enters the Practice Scene and is somewhat familiar with the workspace and components placed in from of him.
2. At first, Saif is not entirely confident in his abilities to interact with the components.
3. This leads to Saif struggling to get accustomed to the environment and objectives.
4. Saif makes mistakes during his attempt to complete the objective.
5. Saif continues to try and complete the objective and in doing so gains valuable knowledge and experience in computer hardware.
6. Saif completes the objective after failing multiple times.

3.2.5 Assessment Scene

Best case scenario:

1. After completing the Practice Scene, Morowa enters the Assessment Scene where she is greeted with familiar components.
2. The system accurately tracks Morowa's movements and interactions, giving him a smooth experience.
3. Morowa successfully assembles the computer within the building scene which increases her confidence.
4. Morowa presses the "Grade" button once she is satisfied with her attempt.
5. Morowa clearly views her customised feedback before navigating the menu option.

Worst case scenario:

1. Kana enters the Assessment Scene but feels the environment is unfamiliar after the help material has been removed.
2. Kana struggles to assemble the components without the help material.
3. Kana does not know how to assemble the components correctly and feels discouraged.
4. Kana presses the "Grade" button to finish her attempt.
5. Kana clearly views her customised feedback before navigating back to the Practice Scene.

Alternative scenario:

1. Timur enters the Assessment Scene after completing the Practice Scene and is presented with the semi-familiar environment.
2. Timur does not feel confident in his abilities to assemble the computer successfully.
3. This leaves Timur confused on his attempt and presses the "Restart" button to clear his score.
4. Timur attempts to assemble the components again.
5. Timur presses the "Grade" button after he is satisfied with his attempt.
6. Timur views his customised feedback but is slightly confused by the grading method.

3.4 Summary

In this chapter, the various use cases and scenarios of the VR software application is discussed. The researcher explored the best-case scenarios that highlighted the power of well-designed VR experiences. The worst-case scenarios brought potential pitfalls and challenges to light, while alternative scenarios showcased different approaches that could influence the functionality of the software.

These discussions emphasised the importance of intuitive interfaces, informative feedback, and seamless functionality in creating immersive and rewarding VR experiences. By considering these use cases and scenarios, VR software can be developed that delivers exceptional user experiences, enhanced learning, and unlocking the full potential of VR technology.

Chapter 4: Technology Used

4.1 Introduction

This chapter focuses on a thorough exploration of the technology that lies the foundation of the development of the proposed application. The complex collaboration among software, coding languages and hardware components requires a thorough comprehension of their individual functionalities as well as their collective interactions. The aspects below will be discussed in terms of the value they present during the development of the proposed VR application.

4.2 Motivation for VR Development Software Used

When selecting the primary software that will be used to develop the VR application, the following aspects should be considered (Baniasadi, Ayyoubzadeh, & Mohammadzadeh, 2020):

- Ease of use – user friendly as well as easy and efficient to use software will minimise development resources in terms of time.
- Learning curve – a lower learning curve can save time during development.
- Platform compatibility – it should support the chosen VR hardware devices like the Oculus Rift, HTC Vive etc.
- Feature set – consider the tools that the VR software offer for designing environments, interactions, and physics handling. The VR software should provide all capabilities needed to successfully develop the application.
- Vendor support – it is important that the software has long term support in terms of updates and patches from the developers.
- Customisation and flexibility – consider whether the software allows the implementation of unique interactions and custom effects or will a third-party plugin become necessary.
- Cost and licensing – some platforms offer free software versions with limited features, while others require a subscription or a one-time purchase.

VR development tools are compared in Table 7 below:

Table 7: VR Software Comparison

(Saleem, 2023)	Unity	Unreal Engine	CryEngine	Maya
Ease of Use	Approachable interface	Advanced capabilities	Robust tools for advanced users	More complex
Learning Curve	Beginner friendly	Steeper learning curve	Moderate learning curve	Professional users
Platform Compatibility	Supports various platforms	Supports various platforms	Primarily PC and VR platforms	Not directly applicable to VR platform
Feature Set	Comprehensive set of tools	Stunning visual capabilities	Impressive graphics	Focus on 3D modeling & animation
Vendor Support	Regular updates & long-term support	Consistent updates	Less frequent updates	Updates focus on improving modeling & animation
Customisation & Flexibility	Customisation through Asset Store & scripting abilities	Extensive customisation	Require technical knowledge to utilize customisation	Highly customizable for modeling & animation
Cost & Licensing	Free & paid options	Free to use with royalty payments for commercial projects	Free to use but licensing required for commercial projects	License purchase

After reviewing the comprehensive comparison of leading VR development software in the above table, the researcher has elected to employ Unity as the platform for developing the proposed application. This choice is rooted in several critical considerations. Firstly, Unity’s cost-free accessibility makes it desirable. Additionally, its seamless compatibility with the chosen hardware ensures a smooth integration process. The inclusion of an extensive array of objects within Unity’s Asset Store further augments its appeal. Lastly, Unity’s expansive developer community, famous for its support, further solidifies this decision.

During the initiation of the project, the researcher opted to employ Unity version 2021.3.20f1 as the development platform. The choice of this specific version was deliberate, as it represented the most recent long-term support (LTS) release available at the project’s outset. LTS versions are known for their stability, reliability, and extended support, making them ideal for projects requiring a solid foundation throughout their development lifecycle.

4.3 Motivation for Modelling Software Used

The decision to use Blender, with a special emphasis on BlenderKit, as the primary tool for modeling computer components within the project is driven by several compelling motivations. BlenderKit, a versatile and accessible resource library within Blender, offers an extensive collection of pre-made 3D models, including a wide range of computer hardware components. Leveraging this resource enabled the acquisition of intricately detailed models of the computer parts, significantly expediting the model creation process.

One advantage of using BlenderKit within the Blender environment is the ability to customise the imported models to suit the unique requirements of the project. Whether it is modifying the dimensions, textures, or materials of the computer components, Blender provided the flexibility and creative control that is needed to ensure the models seamlessly integrate into the VR environment. This approach saves valuable development time and maintains a high level of realism and accuracy in the representation of the computer components.

Finally, Blender’s compatibility with Unity facilitates export capabilities, ensuring that the transition of assets from Blender to Unity occurs smoothly.

4.4 Motivation for VR Hardware Used

The selection of the Oculus Quest as the primary VR platform for my project was motivated by a combination of factors, including the university's existing resources. The university had already invested in Oculus Quest headsets for various purposes, making them readily available for use in this project. This reduces the overall project cost significantly.

Firstly, the Oculus Quest offers an untethered VR experience, allowing users to move freely within the virtual environment without being encumbered by wires or the need for an external PC.

Another compelling aspect of the Oculus Quest is its ease of use and setup. The headset's inside-out tracking system eliminates the need for external sensors, simplifying the installation process.

Additionally, the Oculus Quest boasts the option to interact with the virtual environment using either controllers or their hands. This was a key consideration as it offers versatility in interaction methods as well as an extra layer of immersion.

4.5 Summary

This chapter outlined the deliberate technological choices made for the project. Unity was selected for its user-friendly interface and scripting capabilities, coupled with Oculus Quest for a stand-alone immersive VR experience. Blender played a vital role in crafting lifelike visuals. This chapter emphasizes the importance of version compatibility and long-term support, showcasing the strategic approach followed to focus on delivering an engaging and robust user experience.

Chapter 5: Workspace Design

5.1 Introduction

This chapter is dedicated to the exploration of the deliberate choices in designing the workspace for the VR project. The considerations that guided the decisions will be unveiled, from the spatial arrangement and ergonomic aspects to the incorporation of realistic elements and interactive features (Rebelo, Noriega, Duarte, & Soares, 2012). The workspace, in essence, serves as the canvas upon which users embark on their virtual journey of computer assembly.

5.2 Unity Asset Store

In the creation of the immersive VR experience, a vital resource was the Unity Asset Store, a treasure trove of 3D assets and tools. The environment was meticulously curated by sourcing various objects, textures, and elements from the Unity Asset Store's extensive library (Unity, 2023). Leveraging free downloads available on the platform, high-quality assets were imported that formed the backdrop of the virtual scenes. These assets not only provided visual richness, but also facilitated the seamless construction of the virtual workspace. Each object, carefully chosen and thoughtfully placed, contributed to the immersive quality of the VR experience, enhancing user engagement and educational impact.

5.3 Spatial Layout

VR software design involves the meticulous process of producing engaging and immersive VR experiences for users. It necessitates a multidisciplinary approach that includes user experience design, the creation of an immersive environment, navigation and interactive systems, and input mapping (Cipresso P. , Giglioli, Raya, & Riva, 2018).

It is important to consider the needs and preferences of the users to ensure a seamless and satisfactory experience. To produce such an application, VR software design requires a full understanding of both technical and human-centric factors.

5.3.1 Starting Scene

The inception of any VR experience hinges on the first impression it creates for the user. In pursuit of crafting an immersive and user-friendly VR project, centred around computer assembly, the design of the initial starting scene plays a pivotal role. The starting scene in Figure 13 and Figure 14 was meticulously designed to serve as a gateway into the virtual world.

As users put on the Oculus Quest headset and enter the VR realm, they find themselves in a virtual room that simulates a waiting room. This choice was deliberate, as it provides a familiar and approachable setting.

In this virtual waiting room, a subtle yet visually appealing arrangement of objects, such as a shelf with books (Figure 14), a lamp, and a chair creates an ambiance of comfort and anticipation, setting the tone for what lies ahead.

Figure 13: Starting Scene control wall

However, it is not merely aesthetic that guide the design. Placed strategically within the waiting room is a menu on a flat wall in Figure 13. This is an example of a clear entry point pattern in Human Computer Interaction design. This menu presents users with the choices to ‘Start’ their VR journey or ‘Exit’ if they wish to leave the experience. This user-friendly interface ensures that navigation is straightforward, even for those new to VR.

On the opposite side of the room in Figure 14, a curved wall captures users’ attention. This wall serves a dual purpose: firstly, it hosts essential instructions on how to use the Oculus Quest controllers, ensuring that even novice users can quickly grasp the mechanics of interacting with the VR environment.

Secondly, adjacent to these instructions, a conspicuous cube on a stand invites users to engage and practice interaction. This strategic placement of interactive elements motivates users to explore. Crucially, the curved wall encourages the users' gaze toward the main menu wall behind them. This deliberate design choice ensures that users are naturally oriented towards the starting point of their VR journey.

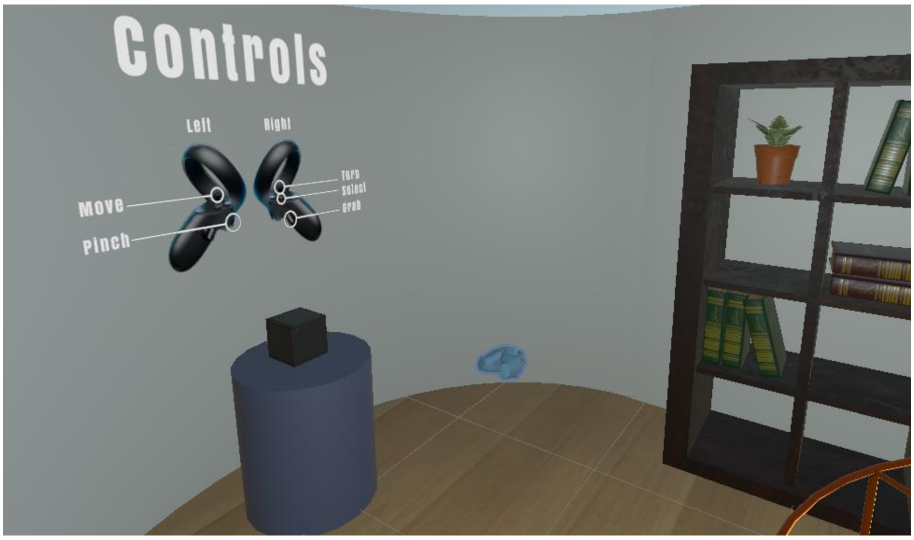


Figure 14: Starting scene menu wall

In essence, the design choices for the starting scene represent the convergence of aesthetics and functionality, ease of use and engagement, familiarity, and innovation. The aim is to provide users with an inviting and instructive introduction to the VR experience, ensuring that from the moment they enter the virtual realm, they feel informed, and eager to embark on the journey of computer assembly.

5.3.2 Building and Assessment Scene

Creating an environment that the user feels immersed in, is crucial for upholding the user experience in a VR application. This emphasises the importance of the environment in setting the tone for the application. As part of a formative assessment, a sample of potential users were given the opportunity to contribute to the design of the environment by voting for their preferred workspace.

Ultimately, the sample of potential users were presented with two options of potential workspaces that could make up the environment of the perceived VR application.

Figure 15 expresses a more creative and colourful workspace that is designed to make the user feel more comfortable and welcomed.

Figure 15: Creative Workspace Design

On the other hand, Figure 16 presents the users with a much cleaner and more professional environment that is designed to keep the focus on the objective of the application. The environment closely resembles a traditional computer lab with minimal objects to distract a user from the task.

Figure 16: Professional Workspace Design

The results of the evaluation of the system indicated that almost 70% of the potential users preferred the professional, computer-lab inspired design, as shown in Figure 16. This option will most likely keep the users focused on the assembly of the computer, which is the desired objective of the application. Any additional objects placed in the workspace that do not directly contribute to the completion of the objective, are included to mimic a realistic experience and to encourage the user to feel comfortable in the workspace.

To avoid confusion, the final iteration of the workspace environment in the Practice and Assessment Scene is similar. In both scenes, a long table stretches against one wall, displaying all the computer components (refer to Figure 17).

Figure 17: Practice Scene

On top of this table rests a high-tech TV screen, which serves as a dynamic informational hub. In the Practice Scene, this screen provides detailed information about each component, offering valuable insights and tips to the user (see Figure 18).

Figure 18: Components Information Panel

Adjacent to the table, on the opposite wall, is an objective panel which is only active in the Practice Scene. This panel serves as a visual guide, displaying the current objectives and goals for the user. It helps users stay on track, ensuring they are aware of their tasks and progress.

Additionally, the objective panel in the Practice Scene offers user-friendly menu options. These options include features like 'Exit' for leaving the practice session, 'Restart' for resetting the current task, and 'Main Menu' for returning to the central hub of the application. They enhance the user's control over their learning experience, allowing them to navigate the Practice Scene seamlessly.

In contrast, the Assessment Scene is deliberately designed to be focused and challenging. The same long table with components remains, providing a familiar setup. However, the informational TV screen now serves a different purpose. In this scene, the screen displays a minimalistic menu, devoid of any guidance or hints (refer to Figure 19).

Figure 19: Assessment Scene Menu Options

Users are encouraged to rely solely on their knowledge and skills, creating an authentic assessment environment. The absence of the objective panel and help materials intensifies the challenge, requiring users to rely on their training and understanding of computer assembly.

Upon successful completion of the Practice Scene, users are greeted with a small congratulatory panel, acknowledging their accomplishment. This panel, shown in Figure 20, provides a sense of achievement and motivation, reinforcing the positive learning experience. Additionally, it provides menu options to exit the game, restart the Practice Scene or go back to the Starting Scene.

Figure 20: Practice Scene Completion Panel

After completing the assessment, users receive detailed feedback on their attempt. Presented in Figure 21, this feedback offers valuable insights into their performance, highlighting areas of improvement and success, ensuring a comprehensive learning and assessment experience.

Figure 21: Assessment Scene Feedback Panel

Together, these design elements create a cohesive and engaging virtual workspace, catering to both learning and assessment needs. The carefully crafted combination of interactive technology and strategic layout ensures that users have a comprehensive and immersive experience, making the process of learning and assessment enjoyable and effective.

5.4 Environment Interaction and Navigation

5.4.1 Hand Presence

The design approach encompassed the implementation of intuitive interaction mechanisms. Opting for Oculus controllers over hand presence or tracking technologies, the researcher prioritised precision, especially for delicate movements involved in hardware assembly. To complement this decision, the researcher integrated modelled hands into the virtual environment, sourced from dedicated packages.

The application's user interface (UI) was optimised through the implementation of Ray Interactors (shown in Figure 22) which is used for interacting with interactable objects at a distance. These interactors were strategically used to simplify interactions with various UI elements. A thoughtful design choice was made to enable Ray Interaction exclusively when users hover over specific UI elements, ensuring a seamless and intuitive navigation experience. This approach not only streamlined the UI, but also enhanced the overall user interaction within the VR environment.

Figure 22: Ray Interactor

5.4.2 Movement in the virtual environment

In the VR application, users navigate and interact using Oculus controllers (see Figure 23). The left controller's joystick enables movement in all directions, while the right controller's joystick facilitates intuitive turning. Users can effortlessly explore the virtual space and inspect components from various angles. Natural head movements further enhance immersion, allowing users to focus on the learning experience. This intuitive control scheme ensures a seamless and enjoyable interaction, enabling users to engage with computer hardware assembly effortlessly.

Figure 23: Oculus Touch Controllers

5.5 Interactivity and User Guidance

The success of a VR application heavily depends on the user’s ability to interact seamlessly with the digital environment. Beyond interaction, effective user guidance is vital for ensuring that users can navigate the VR environment intuitively. In this section, the various facets of interactivity and user guidance that from the backbone of this project, will be discussed.

5.5.1 Interactive Elements in the VR Environment

Central to the VR application's design are the interactive elements that populate the virtual workspace. These encompass a wide array of UI components that contribute to the realistic feel of the environment. These interactive elements are thoughtfully integrated into the VR environment, creating a sense of presence and agency (playing an active role that impacts the virtual world) for the user.

5.5.2 User Guidance and Immersive Learning

In addition to interactive elements, effective user guidance is pivotal in ensuring a smooth and intuitive user experience. Clear visual cues are strategically placed to guide users through the tasks and objectives. The VR environment provides contextual tooltips on components, offering detailed information and instructions, aiding users in understanding the functionalities and placements of each part.

Furthermore, the application integrates immersive learning techniques. Users are encouraged to explore and experiment within the virtual space, fostering a deep understanding of computer hardware components. Gamified challenges and tasks are strategically designed to allow users to progress from fundamental concepts to advanced assembly techniques. Interactive simulations provide users with a risk-free environment to practice and refine their skills, boosting their confidence and competence.

5.5.3 Enhancing Engagement through Feedback

Real-time feedback mechanisms significantly enhance user engagement and learning outcomes. When users successfully assemble components, a subtle visual cue affirm their actions, reinforcing the correctness of their decisions. The que is represented by a green outline that appears around the correct placement when the user hovers the correct object over it as shown in Figure 24.

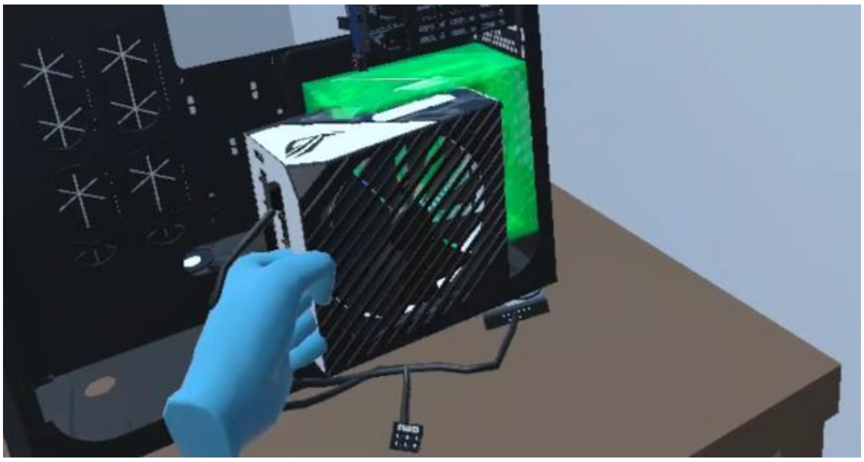


Figure 24: Component Highlighting

Conversely, if users make errors or face challenges, the application provides constructive feedback, guiding them toward the correct solutions. For example, in Practice Mode, the user is not able to assemble the wrong component in the wrong place. The object will fall on release, indicating that the object does not belong in that position. This iterative feedback loop promotes active learning, encouraging users to analyse their mistakes and refine their approach, thereby improving their problem-solving skills.

5.6 Challenges

Designing the VR space for computer assembly came with its challenges. Striking the right balance between guiding users and allowing them to explore independently, was a constant challenge. The goal was to create an environment where users could learn at their own pace, without feeling overwhelmed. This involved experimenting with hints and tooltips to find the right level of assistance. The iterative process led to a dynamic and engaging learning environment.

5.7 Summary

In summary, the careful integration of interactive elements, user guidance strategies, and immersive learning techniques elevates the VR application, creating an engaging and educational experience. By combining interactivity, guidance, and feedback, users are empowered to explore, learn, and master the intricate world of computer hardware assembly within a captivating virtual environment.

Chapter 6: Component Design

6.1 Introduction

In this chapter, the intricacies of the virtual components are explored, investigating the thoughtful design considerations that created the backbone of the VR application. Each element, meticulously crafted and integrated, was aimed at providing users with an authentic and educational experience in computer assembly. From central processing units to intricate cables, every object serves a unique purpose, offering a blend of functionality and visual fidelity.

This chapter is divided into key sections, each dedicated to a specific component crucial to the computer assembly process.

6.2 Component Integration

6.2.1 BlenderKit: A Rich Resource for 3D Assets

BlenderKit is as a vibrant community-driven platform, offering an extensive library of high-quality 3D-assets for artists, developers, and designers. Its vast repository encompasses diverse objects, materials, and textures, all meticulously crafted by skilled artists worldwide as shown in Figure 25. BlenderKit empowers creators by providing ready-made assets, significantly expediting the development process, and ensuring access to top-tier resources.

Figure 25: Computer Components

The following components were collected from the BlenderKit library (BlenderKit, n.d.):

- Made by Joachim Bornemann:
    - Central Processing Unit
    - Graphics Processing Unit
    - Random Access Memory
    - Solid State Drive
    - Power Supply
- Made by Michal CG:
    - Motherboard
    - Computer Case

6.2.2 Tailoring Assets in Blender

Once the assets were procured from BlenderKit, they underwent modifications within Blender, powerful open-source 3D-modelling software. These modifications, for example the cable pins shown in Figure 26, were essential to tailor the assets, ensuring they fit into the virtual workspace. This customisation process not only enhanced visual coherence, but also contributed significantly to the components' intractability.

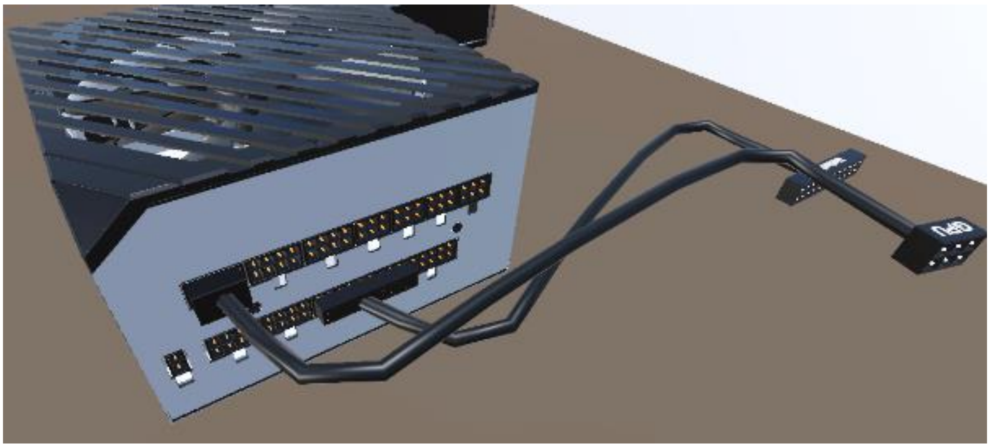


Figure 26: Power Supply with Cabling

6.2.3 BlenderKit Materials and Textures

BlenderKit assets come equipped with high-quality materials and textures, contributing to their realistic appearance. These materials, ranging from metals to plastics, were carefully integrated into the visual environment, enhancing the visual fidelity of the components. By incorporating these meticulously crafted textures, the visual quality of the virtual components was elevated.

6.2.4 Unity Compatibility

BlenderKit assets were not only optimised for Blender, but also tailored for smooth integration into Unity. The researcher adhered to best practices, ensuring that assets were exported with appropriate file formats and scales compatible with Unity. This meticulous preparation facilitated the importation of assets into the Unity development environment.

6.3 Challenges

It takes pride in acknowledging the hurdles that arose during the development of this project. The most daunting challenge that was encountered was the intricacy of setting up the cabling functionality in the application. At the outset, there was confusion, a lack of direction on where to begin, and uncertainty about which functionalities to employ. This struggle demanded not just effort, but ample amounts of time and resilience.

The breakthrough came with the realisation that the cable had to be broken up into different entities, each inheriting from different objects. The starting point of the cable had to inherit from its parent object. The connecting point of the cable had to be all in itself a different object, not inheriting from anything else.

The development of component interactions within Unity was not without its hurdles. The intricate web of Unity's features posed a significant challenge, particularly in understanding the precise configurations of colliders, which is the way Unity handles collisions between objects. Colliders are attached to objects to define the shape of the object for the purpose of physical collisions. Overlapping colliders resulted in disarray, causing connected objects to behave erratically.

Additionally, unravelling the complexities of socket interactors (used for holding interactable objects in a certain position via a socket) demanded a considerable investment of time and effort. Deciphering the ideal colliders for these interactors to seamlessly engage with connecting objects proved to be a puzzle in itself. This journey through Unity's learning curve, though burdensome, was essential. It led to profound insights and a more refined approach, ensuring the final VR application's functionality and user experience.

The countless hours and weeks spent wrestling with these problems are memorable. Each attempt led down a different path, each ending in frustration. These revelations did not come easily. It was the result of tenacity, fueled by the knowledge gained through endless hours of toil and sweat in Unity. The solution came from the inconceivable amount of trial and error, a testament to the tireless determination.

In hindsight, these challenges are celebrated as valuable experiences that contributed to a better understanding of problem-solving. Every moment of confusion and disappointment, every day spent in pursuit of the answer became proof of persistence. The complexity of this project was conquered, and in the end, there was not just a solution but a newfound strength, armed with experience and knowledge gained from relentless pursuit of an answer.

6.4 Summary

This chapter described the intricate world of virtual hardware components, mapping the path from object design to integration. Through meticulous adjustments and strategic integration, the components came to life, creating an immersive, engaging simulation.

Chapter 7: Evaluation

7.1 Introduction

In this chapter, the comprehensive evaluation of the VR computer assembly application is presented. The focus lies on understanding user interactions, identifying improvements, and ensuring the application’s overall effectiveness and user satisfaction.

7.2 Evaluation Overview

In the following sections, the purpose behind the evaluation and the selected evaluation methodologies, along with the rationale behind these choices, will be discussed.

7.2.1 Purpose of Evaluation

The purpose of this evaluation is to assess the usability (specifically effectiveness), and user satisfaction of the VR computer assembly application. Understanding participants’ experiences and challenges is crucial for refining the application, making it intuitive and engaging.

Furthermore, the application can be evaluated from an educational perspective to comprehend its impact on participants’ knowledge and learning outcomes.

7.2.2 Chosen Evaluation Techniques

The application underwent both formative and summative evaluation techniques. These are discussed below.

Formative evaluation

In the formative evaluation process discussed in Chapter 5, a select group of potential participants was engaged in a decision-making process. This step involved presenting them with two distinct design options for the application’s workspace. This deliberate approach aimed to provide the researcher with valuable insights, allowing for a more informed starting point in the application’s development journey.

This formative evaluation technique was carried out over social media via a poll, and a couple of hundred participants responded with their preferred workspace. The results heavily leaned towards a professional, computer lab inspired workspace, with

70% of participants voting for this option. This significantly improved the process of designing the application.

Summative evaluation

In addition to the formative evaluation, a summative evaluation was conducted on the final application. This evaluation aimed to gauge the application's performance against the predefined goals, specifically focusing on three crucial aspects of user experience: user performance, user preference, and user emotions.

7.3 Evaluation Strategy

The evaluation strategy focuses on the target population, the sampling technique employed, and the sample size.

7.3.1 Population

The population chosen for this evaluation comprises individuals interested in computer hardware, ranging from beginners to enthusiasts, ensuring a diverse user base to simulate the range of real-world users.

7.3.2 Sampling Technique

The sampling of convenience method was adopted, selecting participants who are readily available and accessible for the evaluation process. This method was used due to its flexibility and appropriateness for a shorter time period.

7.3.3 Sampling Size

A sample size of 5 participants was chosen, ensuring that there is a balanced representation of the application's potential user base. This sample size is appropriate for the time constraint, and it also follows the "magic number 5" rule, which is the belief that about 80% of usability issues will be observed with the first five participants (Nielsen, 2000)

7.4 Usability Method

The usability test was conducted following a widely recognised user research approach, employing a moderated usability test. In this method, a one-on-one session is facilitated between the moderator (the researcher) and the participant. The moderator guides the participants through the test by posing questions and assigning tasks. The participant's performance and behaviour are recorded, providing valuable data for analysis. The usability test procedure was thoughtfully designed, including specific steps to ensure a thorough and detailed evaluation. These steps are elaborated below:

7.4.1 Usability Test Protocol

The usability test protocol followed a typical format, starting with a pre-test questionnaire incorporating a 5-point Likert scale. This survey aimed to understand the participant's familiarity with VR and computer assembly, serving as valuable insights that could influence their performance and their overall user experience.

Each task was followed by an open-ended question regarding their preferences and dislikes, coupled with an Expectation Measure (used to evaluate how easy or difficult a task is compared to how easy or difficult a participant thought it would be), as part of the post-task ratings. These questions were all verbally answered by the participants and recorded by the moderator.

Following the completion of the pre-test questionnaire, the moderator provided a brief explanation of the application's concept, clarifying the participant's expectations and ensuring a smooth, confusion-free experience. The participants were encouraged to follow the think-aloud protocol, which require them to verbally express their thoughts of the system in real time. This approach facilitated the continuous flow of feedback without interruptions, eliminating the need for participants to remove their headsets for written responses.

Prior to moving on to the tasks, participants were given time to familiarise themselves with the VR headset and controllers, fostering a sense of comfort and confidence.

The starting scene evaluation followed these steps in specific order during the evaluation protocol:

- Participants were asked to identify a specific feature that captured their attention first.
- Additionally, participants were asked to share their initial impressions of the scene, providing valuable qualitative insights captured by the moderator.
- Task 1: This involved interacting with the practice cube. The moderator observed participants' interactions, analysing movements and comments to

assess alignment with expected behaviour. These observations were recorded.

- Task 2: The participants were then presented with their second, and final task, for the starting scene. The moderator instructed the participants to navigate to the Practice Scene. The participants' actions were again observed and analysed for any unexpected behaviour which is noted down.
- During the scene transition, participants were encouraged to express their preferences and dislikes about the environment. An open-ended question encouraged in-depth discussions, allowing participants to discuss their thoughts freely.
- Additionally, the participants answered their Expectation Measure questions.

Once the Practice Scene loaded, the following evaluation protocol was followed step-by-step for the Practice Scene. If participants needed a break from the VR experience, it was provided here:

- Participants were asked about their initial impression upon entering the Practice Scene. This reaction was recorded by the moderator.
- Additionally, the participants were queried about any specific feature or element that caught their attention first, aiding in understanding visual hierarchy and user engagement.
- Task 3: Participants were presented with the main task for this scene: assemble the computer components according to the predefined objectives. Each component assembly was timed, as well as the need for assistance from the moderator. The participants were observed to note any unusual behaviour, responses, and verbal comments.
- Task 4: The final task for the Practice Scene involved participants navigating to the Assessment Scene. Observations were made to take note of any unusual interactions or responses.
- While the Assessment Scene loaded, the participants were asked about their likes and dislikes regarding the Practice Scene. This open-ended question encouraged participants to share their thoughts in detail.
- Additionally, the participants answered their Expectation Measure questions.

Once the Assessment Scene loaded, the following protocol was followed step-by-step for the final Assessment Scene. If participants needed a break from the VR experience, it was taken here:

- Next, participants were encouraged to share their initial impressions, allowing the moderator to understand the immediate impact of the scene’s design.
- Participants were asked to identify the first element that caught their attention upon entering the Assessment Scene.
- Task 5: Participants were given the freedom to explore and assemble computer components in a self-directed manner. During this free play, participants’ interactions were closely observed, noting their strategies, behaviours and challenges faced. This task was timed and the grade that they received was noted for later analysis.
- Task 6: After completing the assembly task, participants were instructed to exit the application while the moderator continued to observe their movements.
- Following the tasks, a discussion was initiated to allow participants to express their preferences, likes, and dislikes about the Assessment Scene.
- Lastly, participants finalised their answers for the Expectation Measure questions.

As the protocol drew to a close and the participants stepped out of the VR environment, they were presented with a final questionnaire which asked if they experienced any cybersickness symptoms and to which extent. They were also asked if this affected their user experience.

This questionnaire additionally contained product reaction cards. Using these cards is an effective way to capture post-test subjective reactions to an application. The final part of the questionnaire contained a simple 5-point Likert scale where the participants rated their computer assembly, VR knowledge, and experience in order to compare it with their initial scores before the start of the usability test. This was to measure the educational value of the application.

7.4.2 Informed Consent Form

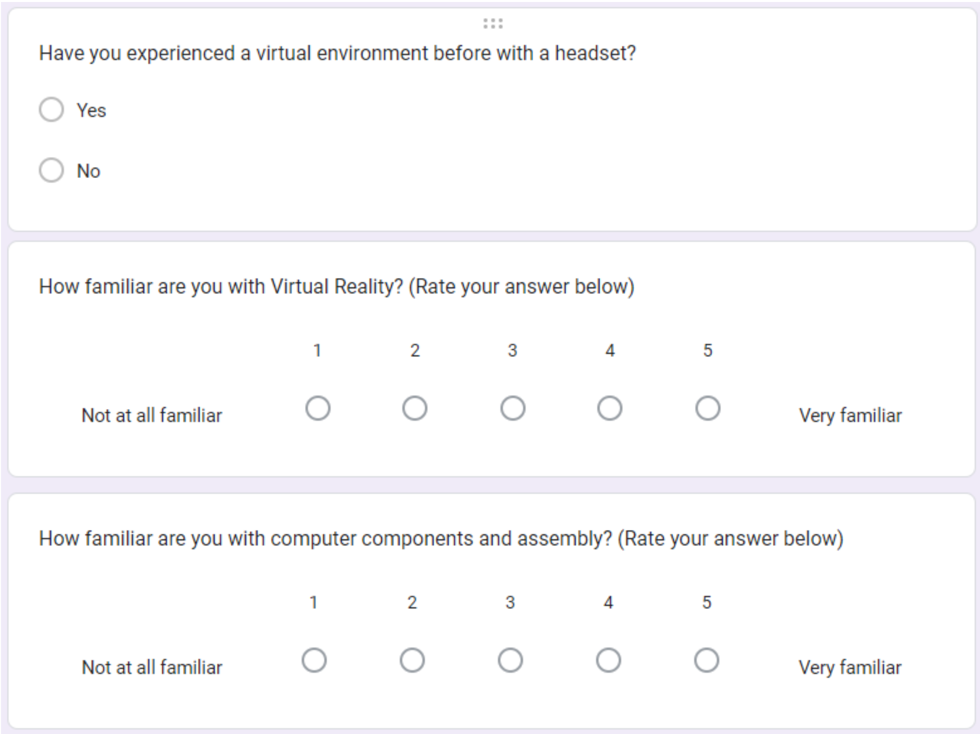
It is important that participants understand the purpose and procedures involved in the usability test before taking part in any tasks. This is encompassed in an informed consent form, shown in Figure 27. They must read and sign this consent form, acknowledging their voluntary participation, understanding of the tasks, and that the recording of their responses would only be for research purposes. Their cooperation ensured valuable insights into the VR application's usability and user experience.

Informed Consent Form	
Title: Usability Test of PC Builder VR **Introduction:** You are invited to participate in a usability test. It is important to read and understand the information provided in this document before you decide to participate in this research test. Please ensure that you read and review this information carefully and understand it fully. Do not hesitate to ask any questions. **Purpose of Research Test:** The purpose of this usability test is to gather information on the usability and user experience of the application, PC Builder VR, to further improve and reflect on the product. **Procedures:** If you agree to participate in this usability test, you will be expected to complete 6 tasks and provide feedback on your experience with the application. Additionally, your response to questionnaires and interview questions will be taken down. The entirety of the study will not take longer than one hour. **Risks and Benefits:** Virtual Reality is known to cause motion sickness in users, therefor there is a high risk of becoming nauseous and/or disoriented during this usability test. However, this study is entirely voluntary, and you have the right to withdraw at any point, especially due to motion sickness. The benefit of this study includes contributing to the research of usability in this application. **Confidentiality:** All information gathered during this usability test will be kept confidential. Only the moderator will have access to the data and any identifying information will be removed before the data is analysed and results are produced. No information will be connected to you in any way. **Voluntary Participation:** Your participation in this usability test is voluntary. You may decide to not participate. You are allowed to withdraw from the test at any point. **Contact Information:** For any questions or concerns about the test and your participation thereof, feel free to contact the moderator, Nadean Barkhuizen via email – 2019602537@ufs4life.ac.za. **Consent:** By signing this form, I acknowledge that I have read and understand the information provided in this form, and that I voluntarily agree to participate in this usability test. I understand that I may withdraw my consent at any time by notifying the moderator.	
Participant Signature:	Date:
Moderator Signature:	Date:

Figure 27: Informed Consent Form

7.4.3 Questionnaires and Interviews

The pre-test questionnaire presented in Figure 28 contained the following questions to gather insight into the participant demographics, which could explain any anomalies. This questionnaire was filled out by each participant after the informed consent form has been signed and their voluntary participation was confirmed.



Have you experienced a virtual environment before with a headset?

○ Yes

○ No

How familiar are you with Virtual Reality? (Rate your answer below)

	1	2	3	4	5	
Not at all familiar	○	○	○	○	○	Very familiar

How familiar are you with computer components and assembly? (Rate your answer below)

	1	2	3	4	5	
Not at all familiar	○	○	○	○	○	Very familiar

Figure 28: Pre-Test Questionnaires

Next, the observation sheet, as shown in Figure 29, was used during the usability test. All statements and answers were recorded by the moderator to ensure a smooth process. However, it must be noted that the participants might feel uncomfortable to express negative sentiment towards the application. This is called the social desirability bias in which participants tend to give answers they believe will make them look better in the eyes of the moderator (Nancarrow & Brace, 2000).

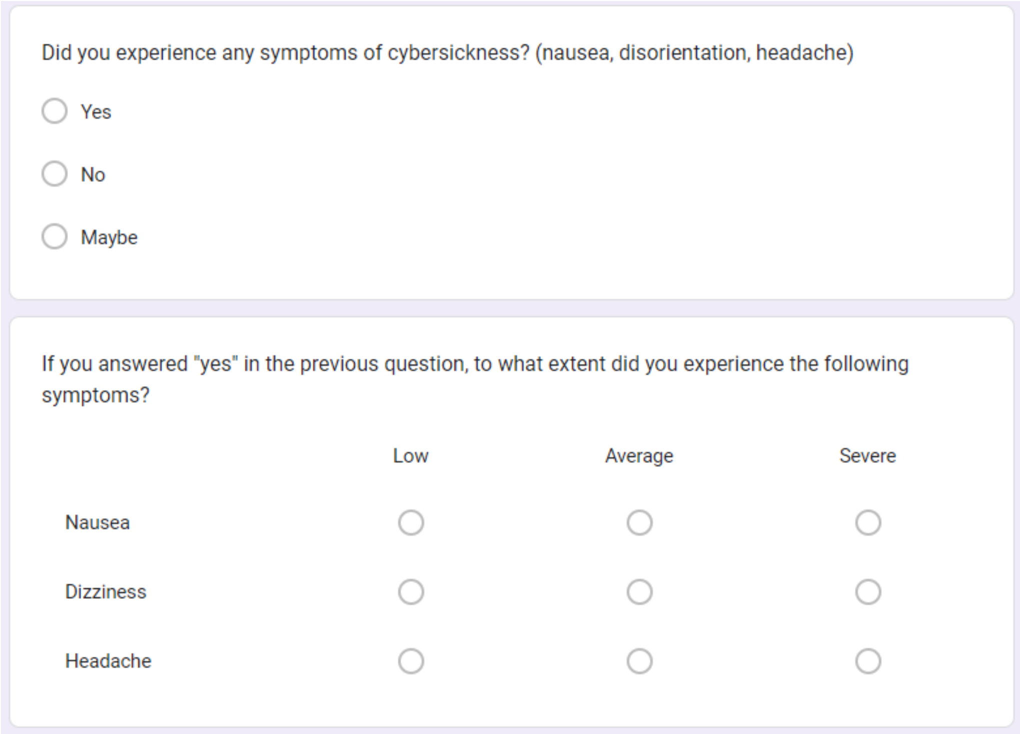
Observation Sheet	
Starting Scene:	**Answers & Notes:**
Element noticed first:	
Initial impression:	
Task 1: Interaction with test cube	
Expectation rating:	◻ 1 ◻ 2 ◻ 3 ◻ 4 ◻ 5 ◻ 6 ◻ 7

Task Level Success:	Complete Success: □ With Assistance □ Without Partial Success: □ With Assistance □ Without Failure: □ Not complete □ Gave up
Notes on interaction:	
Participant comments made:	
Experience rating:	□ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7
Task 2: Navigate to Practice Scene	
Expectation rating:	□ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7
Task Level Success:	Complete Success: □ With Assistance □ Without Partial Success: □ With Assistance □ Without Failure: □ Not complete □ Gave up
Time on task:	___________ minutes ___________ seconds
Participant likes:	
Participant dislikes:	
Participant comments made:	
Experience rating:	□ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7
Practice Scene:	**Answers and Notes:**
Element noticed first:	
Initial impression:	
Task 3: Assemble components	
Expectation rating:	□ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7
Task Level Success:	Complete Success: □ With Assistance □ Without Partial Success: □ With Assistance □ Without Failure: □ Not complete □ Gave up
Time on task for component:	1. __________ seconds Cable 1. __________ sec 2. __________ seconds Cable 2. __________ sec 3. __________ seconds Cable 3. __________ sec 4. __________ seconds 5. __________ seconds 6. __________ seconds 7. __________ seconds 8. __________ seconds 9. __________ seconds

	Total time to complete: ________ min _______ sec
Comments made:	
Experience rating:	□ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7
Task 4: Navigate to Assessment Scene	
Expectation rating:	□ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7
Task Level Success:	Complete Success: □ With Assistance □ Without Partial Success: □ With Assistance □ Without Failure: □ Not complete □ Gave up
Time on task:	__________ minutes __________ seconds
Participant likes:	
Participant dislikes:	
Participant comments made:	
Experience rating:	□ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7
Assessment Scene	**Answers and Notes:**
Element noticed first:	
Initial impression:	
Task 5: Free Play Grade	
Expectation rating:	□ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7
Time on task:	__________ minutes __________ seconds
Grade:	__________ %
Participant comments made:	
Experience rating:	□ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7
Task 6: Exit Application	
Expectation rating:	□ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7
Task Level Success:	Complete Success: □ With Assistance □ Without Partial Success: □ With Assistance □ Without Failure: □ Not complete □ Gave up
Time on task:	__________ minutes __________ seconds
Participant likes:	
Participant dislikes:	
Experience rating:	□ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7

Figure 29: Observation Sheet

After they completed all the tasks, the participants were presented with a closing questionnaire, asking if they experienced cybersickness, and to what extent. The questionnaire can be found in Figure 30.


Did you experience any symptoms of cybersickness? (nausea, disorientation, headache)

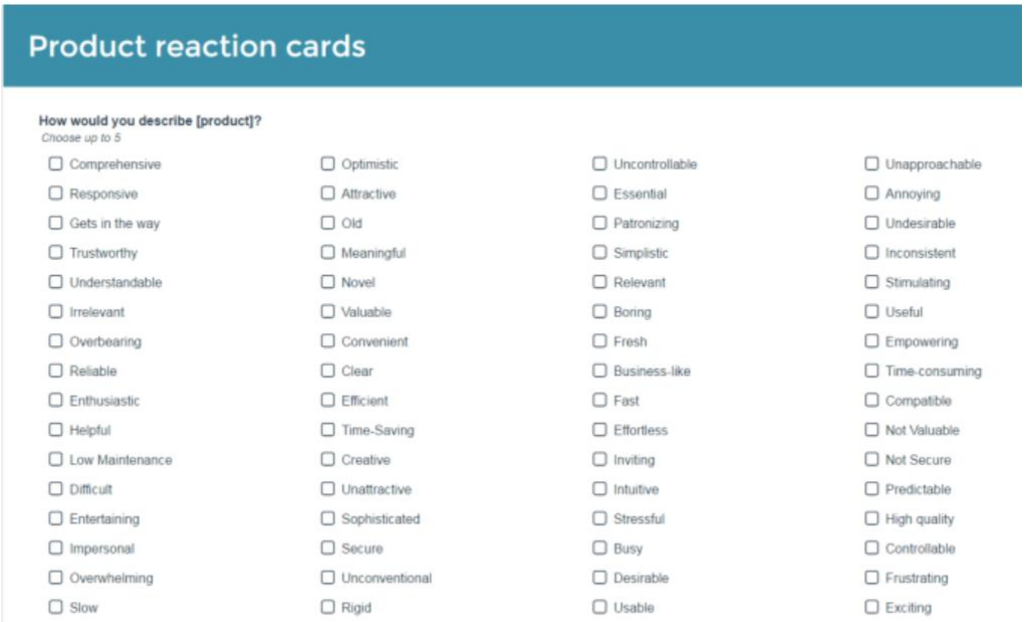
- Yes
- No
- Maybe

If you answered "yes" in the previous question, to what extent did you experience the following symptoms?

	Low	Average	Severe
Nausea	○	○	○
Dizziness	○	○	○
Headache	○	○	○

Figure 30: Post-Test Questionnaire

The product reaction cards were presented in a table format for quick identification. The table is presented in the similar format as in Figure 31 below:

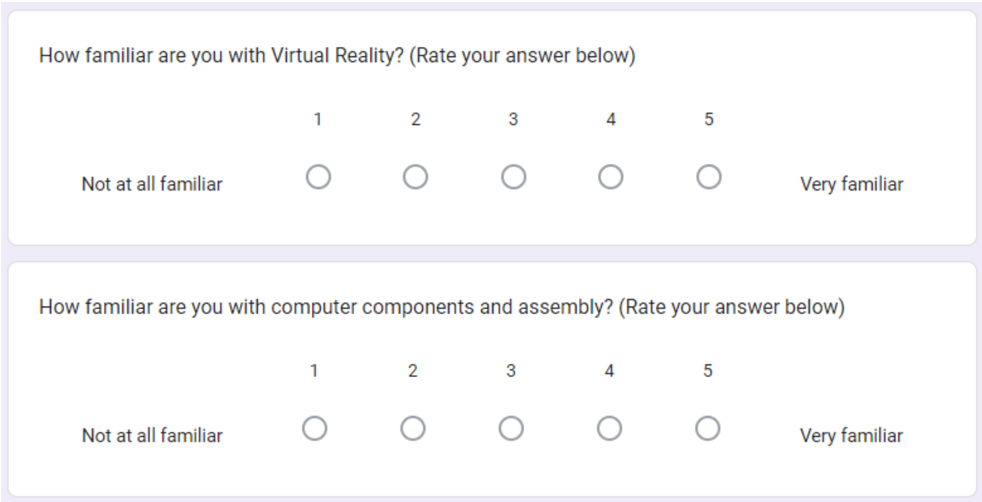

Product reaction cards

How would you describe [product]?
Choose up to 5

Comprehensive	Optimistic	Uncontrollable	Unapproachable
Responsive	Attractive	Essential	Annoying
Gets in the way	Old	Patronizing	Undesirable
Trustworthy	Meaningful	Simplistic	Inconsistent
Understandable	Novel	Relevant	Stimulating
Irrelevant	Valuable	Boring	Useful
Overbearing	Convenient	Fresh	Empowering
Reliable	Clear	Business-like	Time-consuming
Enthusiastic	Efficient	Fast	Compatible
Helpful	Time-Saving	Effortless	Not Valuable
Low Maintenance	Creative	Inviting	Not Secure
Difficult	Unattractive	Intuitive	Predictable
Entertaining	Sophisticated	Stressful	High quality
Impersonal	Secure	Busy	Controllable
Overwhelming	Unconventional	Desirable	Frustrating
Slow	Rigid	Usable	Exciting

Figure 31: Product Reaction Cards (Turner, 2016)

The final part of the post-test evaluation consisted of the same questions that the participants were asked to answer as part of the pre-test questionnaire (see Figure 32). This was done in an attempt to assess the application's ability to act as an educational tool to teach participants about computer components and assembly.


How familiar are you with Virtual Reality? (Rate your answer below)

	1	2	3	4	5	
Not at all familiar	○	○	○	○	○	Very familiar

How familiar are you with computer components and assembly? (Rate your answer below)

	1	2	3	4	5	
Not at all familiar	○	○	○	○	○	Very familiar

Figure 32: Post-Test Comparison Questionnaire

7.4.4 Metrics Collected

The following metrics were collected during the usability test:

- Performance metrics:
  - Task success was captured by measuring levels of success.
  - Time-on-task was captured by timing the participants.
  - Errors were collected by observing the participants and their behaviour within the virtual environment. Any strange reactions or interactions were recorded as errors if it deviated from the direct route to completing the task at hand. For example, pressing the wrong button.
- Self-reported metrics:
  - The participants were tasked to rate certain aspects of themselves or the application on a rating scale.
  - The participants were asked to complete post-task ratings via Expectation Measure questions.
- Issue-based metrics:
  - Participants were asked to use the think-aloud protocol during their study and the moderator made notes to highlight potential usability issues.
  - Participants' verbatim comments were also noted if it revealed an issue.

7.5 Analysis

In this section, the data collected from the usability test is analysed to extract key insights into user interactions and feedback. Examining participant demographics, task success rates, and user preferences, the aim was to comprehensively understand the application's usability and overall user experience.

7.5.1 Participant Demographic Information

The collected demographic data shed light on the participants' backgrounds before they engaged in the usability test.

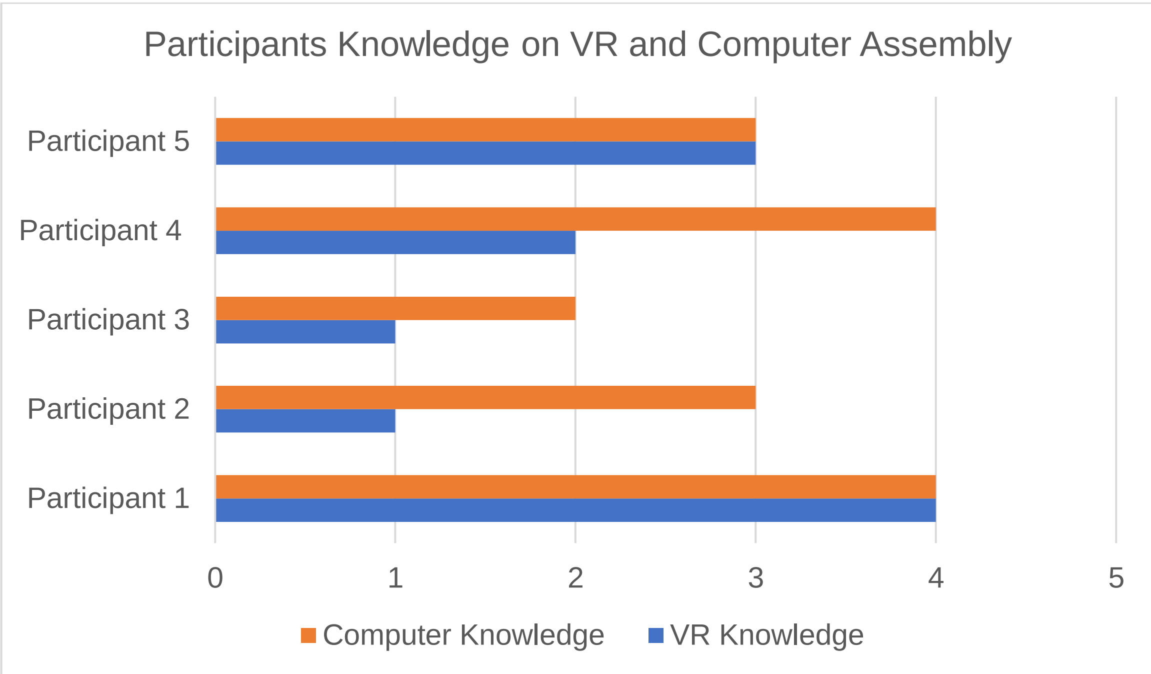


Figure 33: Participants Knowledge on VR and Computer Assembly

Referring to Figure 33, participant number 1 emerged as the most adept in both VR and computer assembly, setting a promising tone for the study. The average across all participants revealed the following:

- On average, the participants demonstrated a basic understanding, with a mean computer knowledge score of 3.2 out of 5. This aligns well with the expected proficiency level of the target audience.
- The average level of VR knowledge for the participants was 2.2 on a 5-point rating scale. This number is very reasonable, since VR is not a very accessible technology.

Only one participant had prior experience with VR (see Figure 34), which notably influenced the higher rating of 4 out of 5 in VR knowledge.

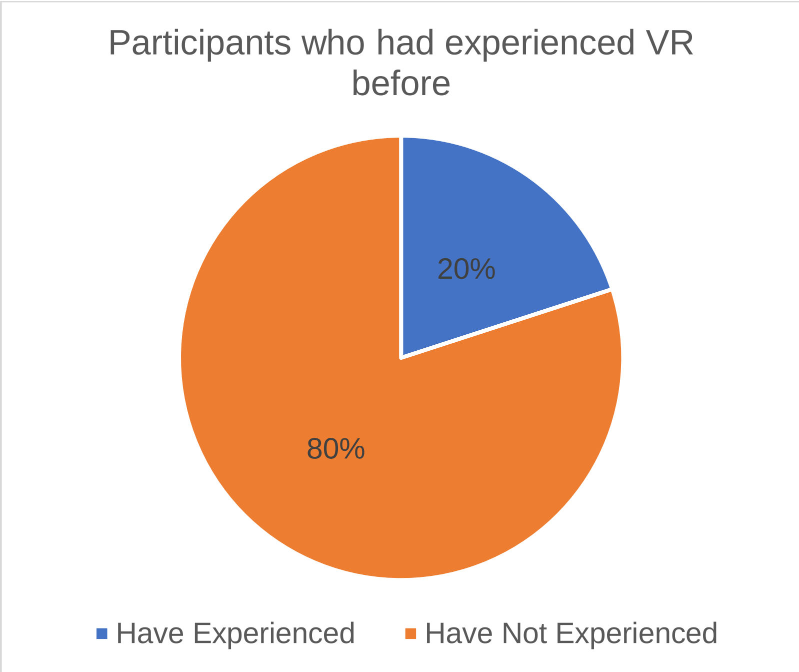


Figure 34: Participants who had Experienced VR Before

The participant demographics did not reveal any anomalies and the results are in line with what the researcher expected. These demographic results will be revisited for a post-test comparison.

7.5.2 Analysis of Metrics

The following metrics were recorded, and the analysis thereof is discussed per task below.

Starting Scene:

The participants were asked to specify which element first grabbed their attention. However, since the starting scene was designed in a very specific way, all participants noticed the controller instructions on the wall first.

Additionally, they were asked to report on their initial impression of the environment. All participants expressed fascination with the environment, especially the bookcase, chair, and lighting. The reason for this was that most participants have never experienced VR before.

Task 1: The participants were observed as they interacted with the test cube. All participants navigated this task with relative ease, despite some initial clumsiness. They intuitively looked at the controls for guidance and successfully picked up the cube without any notable issues. Their ability to swiftly comprehend the controller functions showcased a basic, yet promising, level of proficiency.

No errors were recorded during this task. The researcher did not classify pressing the wrong button to pick up an object as an error, unless the participant repeatedly attempted to use the same incorrect button without exploring other options. Such occasional button missteps did not hinder task completion and were not considered task failures. Therefore, this task revealed a 100% success rate, which showed promise for the rest of the application.

Task 2: The participants were observed while they were required to navigate to the Practice Scene from the starting environment. All participant intuitively used the joysticks to move within the environment and face the menu wall. With no assistance from the moderator, all participants successfully completed the task, revealing a 100% task success rate for the second task. No task failures were observed during this stage.

At this point the participants discussed their likes and dislikes of the starting scene, which revealed the following:

- Participants expressed a liking towards the elements placed within the environment, such as the chair, bookcase, and lighting. They elaborated that this made for a comfortable and familiar environment to be immersed in.
- Very few negative sentiments emerged in this scene, which could be blamed on the social desirability bias. However, one participant expressed that the cube on the stand was too close to the wall that displayed the controls.

Practice Scene:

The angle that participants faced when entering the Practice Scene, allowed them to face all elements of interest in the environment. Two out of five participants revealed that they noticed the objective panel on the left-hand side of the environment, while the rest of them noted that they observed the table with components on the right-hand side of the environment first.

The participants' initial response to the environment was positive; they expressed enthusiasm for its realistic design. They particularly appreciated the clear presentation of objectives, and the cohesive arrangement of all elements in the scene.

Task 3: Participants were challenged to assemble the computer components within the Practice Scene by following the objectives. While the task proved to be a moderate challenge, with 4 out of 5 participants successfully completing it with assistance from the moderator, one participant did not require assistance and used the helpful resources as a guide. Notably, the participant who completed the task independently, had a higher rating of 4 for computer assembly knowledge.

The average completion time for the task was 7 minutes and 41 seconds, indicating a steady pace among the participants. This time duration is lower than what the researcher expected. However, it is highly likely due to the moderator’s assistance during task performance.

Task 4: The participants were observed while they were required to navigate to the Assessment Scene from the practice environment. All participant intuitively used the controllers to interact with the presented UI panel to navigate the system. With no assistance from the moderator, all participants successfully completed the task, revealing a 100% task success rate for the fourth task. No task failures were observed during this stage.

At this point the participants discussed their likes and dislikes of the Practice Scene, which revealed the following:

- All participants expressed that they owed their success in completing the task to the functionalities in the application that were designed to help the participant during assembly. The visual confirmation that they correctly placed a component heavily aided them with their attempt.
- All participants encountered some level of difficulty with cabling due to its small size, making it challenging to observe and manipulate accurately within the environment.

Overall, the participants were very surprised and in favour of the visual cues that aided them in the assembly, for example, the green highlighting of correct component placement. This functionality is highly valuable in affirming that participants were doing the right thing, which motivated them to continue.

Assessment Scene:

The Assessment Scene deliberately followed the exact same design for the environment as for the Practice Scene. All participants expressed appreciation for this choice, and they noted that the scene felt familiar. It also enhanced the learnability of the application.

When participants were prompted to discuss which element they noticed first, all five participants responded that they noticed that the objective panel has been removed from the environment. Additionally, they noticed the menu options that were available on the right-hand wall. These reactions were welcomed and expected due to the deliberate choice of design, and the removal of help material.

Task 5: The participants were observed while they explored the environment freely and assembled the computer to their standard. Unlike previous tasks, errors related to incorrect assembly, or the need for assistance, were not applicable here. The only potential point of error could occur when participants selected the 'Grade' button. However, all participants successfully completed this task without any issues. All participants assembled the computer and graded their attempts, as can be seen in Figure 35.

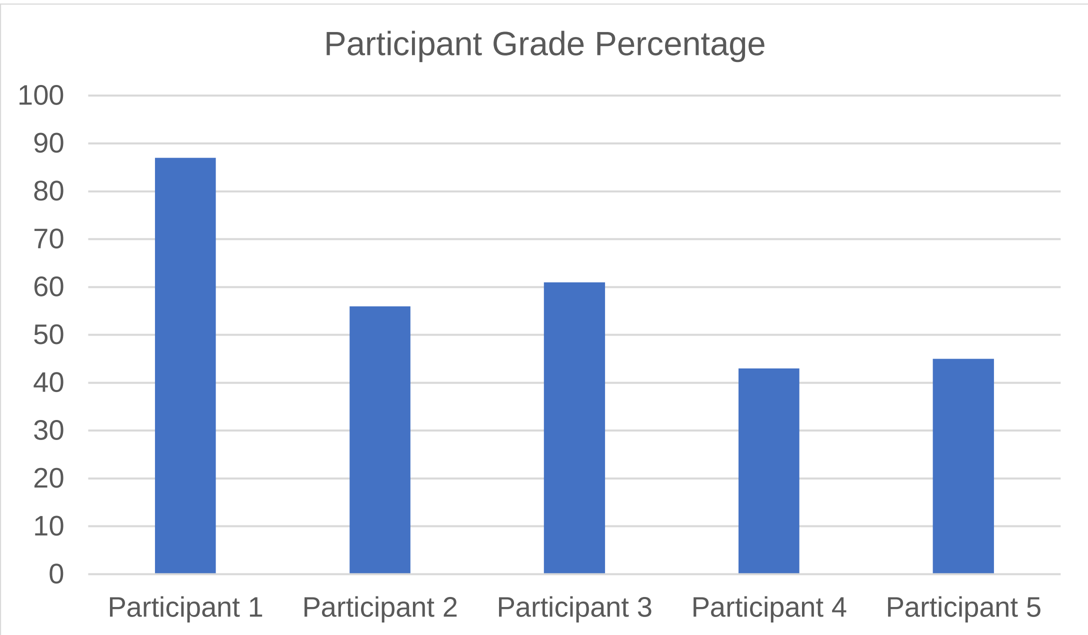


Figure 35: Participants' Grade Percentage

The average grade received by participants was 58.4%. This grade, although relatively high for inexperienced participants, contrasted with the incorrect placements observed. Strategies to address this discrepancy are outlined in the suggested changes section.

The average completion time for this task was 10 minutes and 31 seconds, which exceeded the time taken in the practice round. When participants were asked about the extended duration, they attributed it to the valuable assistance provided by the help materials. According to their feedback, these resources significantly aided them in successfully assembling the computer.

Task 6: All participants successfully carried out the task of exiting the application, with an impressive average time of 5 seconds. They attributed this time to the button being effectively placed directly in their point of view. Their efficient navigation highlighted their familiarity with the VR environment, demonstrating a smooth exit process.

At this point the participants discussed their likes and dislikes of the Practice Scene, which revealed the following:

- The participants expressed a liking towards the familiar environment. They felt comfortable and confident within the space.
- The participants mentioned that the assembly process without help material proved to be more challenging than expected. However, they revealed that with more practice attempts, the assessment process may become easier.

7.5.3 Analysis of Post-Test Questionnaire/Interview

Pre-/Post-Task Questionnaire

The pre-/post-task questionnaire was administered during the usability test. Before each task, the participant was asked to rate (on a 7-point Likert scale) how easy/difficult they expect the task to be. After the task was completed, they were asked to rate how easy/difficult they experienced the task to be, on the same scale.

This is called an Expectation Measure and was used to determine the participant’s subjective reaction to each task. This information is highly valuable in determining where to make improvements in the design of the product.

Table 8 below contains all the *expectation* measures for all 6 tasks. The right-hand column contains the averages, which will be plotted on a scatterplot diagram for analysis (see Figure 36).

Table 8: Expectation Measure

Expectation Measure						
	Participant Number					
Task	**1**	**2**	**3**	**4**	**5**	**Average**
1	7	4	3	2	3	3.6
2	7	3	4	3	4	4.2
3	4	2	2	2	3	2.6
4	7	4	4	5	4	4.8
5	3	2	1	1	2	1.8
6	7	6	6	6	6	6.2

Table 9 contains all the *experience* measure for all 6 tasks. The last column of the table contains the averages, which will be plotted and analysed, along with the expectation measures, on the scatter graph in Figure 36 below.

Table 9: Experience Measure

Experience Measure						
	Participant Number					
Task	**1**	**2**	**3**	**4**	**5**	**Average**
1	7	5	6	5	4	5.4
2	7	5	4	4	6	5.2
3	6	5	4	4	5	4.8
4	7	6	6	7	7	6.6
5	5	4	3	4	4	4
6	7	7	7	7	7	7

The scatter plot in Figure 36 revealed the following results:

- Task 2, 4 and 6 (all being the navigation tasks) fell into the “Don’t Touch It” quadrant, which means that the participants thought the task would be easy, and it turned out to be easy. You do not want to touch it, because you do not want to “break it”.

- Task 1, 3 and 5 (being the interactive tasks) fell into the “Promote It!” quadrant, which means that participants thought they would be difficult to complete, but they actually turned out to be easy to complete. This distinguishes the system from other systems and should be “promoted”.

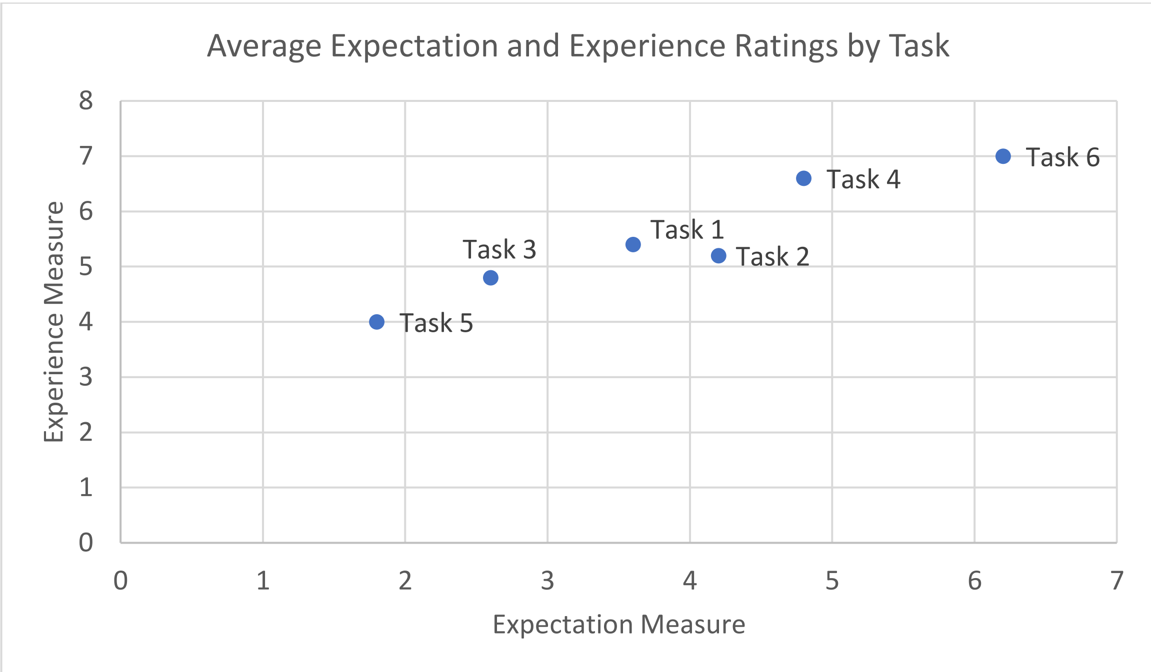


Figure 36: Average Expectation and Experience Ratings by Task

Product Reaction Cards

The product reaction cards’ results were analysed and placed into a word cloud to easily visualise the results (refer to Figure 37). The word cloud revealed a highly positive sentiment towards the application, with frequent occurring words like “Helpful”, “Valuable” and “Exciting”. However, some negative sentiment, like “Stressful” and “Overwhelming”, was also noted. The participants expressed that the concept of computer assembly in a virtual environment, was causing a nervous response. However, the fact that the participants were placed in a usability test, could account for this effect.

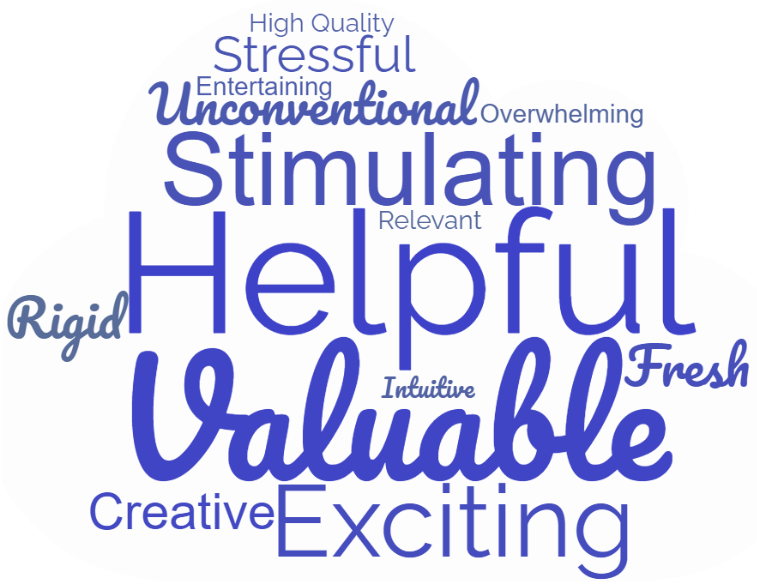


Figure 37: Word Cloud from Product Reaction Cards

Cybersickness questionnaire

The results from the cybersickness questionnaire revealed noteworthy findings. Among the participants, four out of five reported experiencing various degrees of cybersickness symptoms. Strikingly, two participants disclosed severe symptoms, including nausea and dizziness. This adverse impact on their overall user experience left a significant impression. However, after reassurance that cybersickness tends to reduce with continued exposure and understanding, these individuals expressed a tentative willingness to give VR another try, showcasing the potential for positive change in future engagements.

Post-test Learnability Questionnaire

The post-test questionnaire on the participants' computer assembly and VR knowledge showed significant improvement (as can be seen in Figure 38), with computer assembly knowledge increasing with 12%, and VR knowledge increasing with 28%. This meant that the participants viewed the application as a tool to improve these two skillsets.

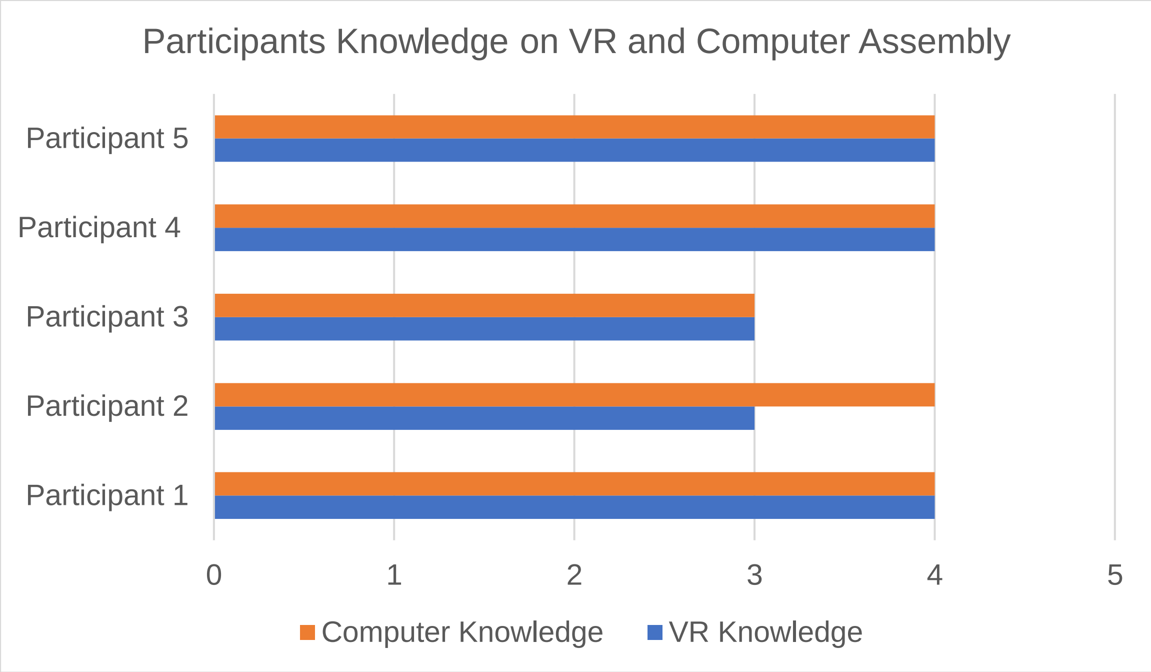


Figure 38: Participants Post-Test Knowledge on VR Computer Assembly

7.6 Summary of Suggested Changes

In response to user feedback and in consideration of minimising the cybersickness risk, several strategic changes were proposed for the VR application:

1. Grading System Revision: A stricter grading system should be implemented, ensuring a balanced and challenging assessment criterion, thus fostering a sense of accomplishment without compromising difficulty.

2. Enhanced Cabling Functionality: To alleviate user frustration during the cabling phase, a new feature must be introduced. Upon entering the cabling stage, an option should be available to enlarge cables, enhancing visibility and ease of manipulation, especially for users with limited experience.

3. Teleportation Functionality: To reduce the need for extensive physical movement and mitigating the cybersickness risk, a teleportation functionality could replace the walking mechanism. Teleportation is a common and user-friendly navigation method in VR applications, providing users with precise control over their movement without inducing discomfort.

By integrating these changes, it is expected that the VR experience will be more accessible, enjoyable, and inclusive, promoting a positive learning environment for all participants.

7.7 Summary

In this chapter, the VR application underwent exploration by participants, providing valuable insights into user interactions, challenges faced, and overall user experience.

Through a systematic usability test and in-depth analysis, participants' reactions, preferences, and performance were carefully examined. These findings served as a foundation for refining the VR application, ensuring a seamless and user-friendly experience, ultimately enhancing its educational value and user satisfaction.

Chapter 8: Source Code and Implementation

8.1 Introduction

In this chapter, the source code of the VR application, specifically the intricacies of the codebase that powers the immersive experience, is explored. Insights can be gained into the scripting techniques, solutions, and integration of this application.

8.2 Unity Source Code

In Unity, scripting is the process of creating custom behaviours and interactions for game objects within the project. It facilitates C# for this purpose and follows an Object-Oriented Programming approach. In Unity, everything is a GameObject, which is an empty container that can hold components. Components are scripts or functionalities that can be added to GameObjects.

Unity scripts can access and modify these components. Unity calls specific methods in the script at various points in the object's lifecycle. These functions can be overridden to implement custom behaviour. Scripts can access and modify component properties.

The MonoBehaviour class, which serves as Unity's base class for components, is the class that any new C# class that you create, automatically inherits from (Kromenaker, 2013).

Hand Animation

The 'HandAnimatorController' (Figure 39) plays a crucial role in the VR experience, translating users' hand movements into real-time visual feedback on their controllers. This mapping ensures a seamless connection between user actions and on-screen interactions, enhancing immersion and providing accurate, responsive feedback throughout the VR experience.

This script integrates Unity's Input System to synchronise VR controller input with virtual hand animations. It reads trigger and grip button presses, mapping them to corresponding animation parameters, enabling realistic hand movements within the VR environment.

```csharp
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.InputSystem;

public class HandAnimatorController : MonoBehaviour
{
    [SerializeField] private InputActionProperty triggerAction;
    [SerializeField] private InputActionProperty gripAction;

    private Animator animator;

    // Start is called at first frame
    private void Start()
    {
        animator = GetComponent<Animator>();
    }

    // Update is called once per frame
    private void Update()
    {
        float triggerValue = triggerAction.action.ReadValue<float>();
        float gripValue = gripAction.action.ReadValue<float>();

        animator.SetFloat("Trigger", triggerValue);
        animator.SetFloat("Grip", gripValue);
    }
}
```

Figure 39: Hand Animator Controller Script

Scene Changes

```csharp
using UnityEngine;
using UnityEngine.SceneManagement;

public class SceneController : MonoBehaviour
{
    public FeedbackManager feedbackManager;
    public AssessmentTracker assessmentTracker;
    public GameObject feedbackPanel;
    public GameObject[] objectsToDisable;

    public void RestartLevel()
    {
        SceneManager.LoadScene("AssessmentScene");
    }

    public void GoToMainMenu()
    {
        SceneManager.LoadScene("StartingScene");
    }

    public void GradeAttempt()
    {
        feedbackManager.UpdateUIFeedback(assessmentTracker);
        feedbackPanel.SetActive(true);

        foreach (GameObject objects in objectsToDisable)
        {
            objects.SetActive(false);
        }
    }

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

Figure 40: Scene Controller Script

The ‘SceneController’ script (Figure 40) leverages Unity’s Scene Manager, a tool for handling scene changes within the application. This script is assigned to an empty GameObject, and the methods are linked to buttons used to navigate the system.

Practice Scene Components

The 'CPUSnapping' script (Figure 41) , integrated into each interactable component in the Practice Scene, facilitates the update of objectives during assembly. When the component becomes active at run-time, a listener is added to the 'selectEntered' event of the component's destination socket interactor. This event is triggered when a component is inserted into the socket. This event calls certain methods that lead to the updating of the component's objective on the panel.

```
using UnityEngine;
using UnityEngine.Events;
using UnityEngine.XR.Interaction.Toolkit;

public class CPUSnapping : MonoBehaviour
{
    public XRSocketInteractor socketInteractor;
    public ObjectiveTracker objectiveTracker;

    private void OnEnable()
    {
        socketInteractor.selectEntered.AddListener(OnCPUInserted);
        socketInteractor.selectExited.AddListener(OnCPURemoved);
    }

    private void OnDisable()
    {
        socketInteractor.selectEntered.RemoveListener(OnCPUInserted);
        socketInteractor.selectExited.RemoveListener(OnCPURemoved);
    }

    private void OnCPUInserted(SelectEnterEventArgs args)
    {
        XRBaseInteractable interactable = args.interactableObject as
        XRBaseInteractable;
        if (interactable != null)
        {
            objectiveTracker.UpdateObjective("MountCPU");
        }
    }

    private void OnCPURemoved(SelectExitEventArgs args)
    {
        XRBaseInteractable interactable = args.interactableObject as
        XRBaseInteractable;

        if (interactable != null)
        {
            objectiveTracker.HandleComponentRemoval("MountCPU");
        }
    }
}
```

Figure 41: CPU Snapping Script

Objective Tracker for Practice Scene

The 'Objective Tracker' class (Figure 42) oversees the management of objectives within the Practice Scene. Defined in the class are objectives maintained in a list, each with a unique name and completion status. During runtime, the class maps the objectives to UI text elements for display purposes. The class contains updating methods for the objective based on user actions like component assembly or removal. It also dynamically updates UI text in the environment based on these status changes.

Once all objectives are completed, a congratulatory message is displayed, and the user can navigate the system further.

```
public class ObjectiveTracker : MonoBehaviour
{
    [System.Serializable]
    public class Objective
    {
        public string objectiveName;
        public bool isCompleted;
    }

    public List<Objective> objectives = new List<Objective>();
    // List of game objects & UI text

    void Start()
    {
        objectiveTextDictionary["MountCPU"] = cpuText;
        objectiveTextDictionary["MountGPU"] = gpuText;
        // Mappings for other objectives
    }

    public void UpdateObjective(string objectiveName)
    {
        Objective objective = objectives.Find(obj => obj.objectiveName ==
        objectiveName);

        if (objective != null)
        {
            // Call update methods
        }
    }

    private void UpdateTextStrikethrough(string objectiveName, bool isCompleted)
    {
        if (objectiveTextDictionary.TryGetValue(objectiveName, out TextMeshProUGUI
        textToUpdate))
        {
            if (textToUpdate != null)
                if (isCompleted)
                    textToUpdate.text = $"<s>{textToUpdate.text}</s>";
                else
                    textToUpdate.text = textToUpdate.text.Replace("<s>", "").Replace("</s>", "");
                // Change text transparency
|       }

    public void HandleComponentRemoval(string componentName)
    {
        Objective objective = objectives.Find(obj => obj.objectiveName == componentName);
        if (objective != null)
            objective.isCompleted = false;
            UpdateTextStrikethrough(componentName, false);
    }

    private void CheckAllObjectivesCompleted()
    {
        bool allCompleted = objectives.TrueForAll(obj => obj.isCompleted);
        if (allCompleted)
        {
            // Enables panel that displays congratulatory message and offers navigation
        }
    }
}
```

Figure 42: Objective Tracker Script

Assessment Tracker for Assessment Scene

This class (Figure 43) manages the assessment process within the application. It maintains dictionaries that track the correct placement status and the count of incorrect placements for the components. If a component is incorrectly placed, its incorrect placement counter is incremented. This class is used to retrieve the placement status and incorrect placement count for another script to utilise to present feedback to the user on their assembly attempt.

The Awake() method is used to implement the Singleton pattern. This ensures that the class has only one instance and provides a global point of access to that instance. In the Awake() method, it checks if the _instance variable is null. If another instance of the ‘AssessmentTracker’ is created, for example when the scene is restarted, the existing instance is destroyed and the new instance is assigned to the variable.

```
using UnityEngine;
using System.Collections.Generic;
using UnityEngine.XR.Interaction.Toolkit;

public class AssessmentTracker : MonoBehaviour
{
    private Dictionary<XRBaseInteractable, int> incorrectPlacementCount = new
    Dictionary<XRBaseInteractable, int>();
    private Dictionary<XRBaseInteractable, bool> interactablesDictionary = new
    Dictionary<XRBaseInteractable, bool>();

    private static AssessmentTracker _instance;

    private void Awake()
    {
        if (_instance == null)
            _instance = this;
        else
            Destroy(gameObject);
    }

    public void RegisterInteractable(XRBaseInteractable interactable)
    {
        incorrectPlacementCount[interactable] = 0;
        interactablesDictionary[interactable] = false;
    }

    public void HandlePlacement(XRBaseInteractable interactable, bool isCorrectPlacement)
    {
        if (interactablesDictionary.ContainsKey(interactable))
            interactablesDictionary[interactable] = isCorrectPlacement;

            if (!isCorrectPlacement)
                if (incorrectPlacementCount.ContainsKey(interactable))
                    incorrectPlacementCount[interactable]++;
    }

    public bool GetPlacementStatus (XRBaseInteractable interactable)
    {
        if (interactablesDictionary.ContainsKey(interactable))
            return interactablesDictionary[interactable];
        else
            return false;
    }

    public int GetIncorrectPlacementCount(XRBaseInteractable interactable)
    {
        if (incorrectPlacementCount.ContainsKey(interactable))
            return incorrectPlacementCount[interactable];
        else
            return 0;
    }
}
```

Figure 43: Assessment Tracker Script

8.3 Summary

In summary, the integrated selected scripts, presented in this chapter, are fundamental to the VR application's seamless functionality. They enable precise hand movement mapping, facilitate scene transitions, provide interactive elements, and record user interactions for feedback. These scripts collectively enhance the user experience, ensuring an engaging and immersive environment for learning.

Chapter 9: Challenges Faced

9.1 Introduction

Developing a comprehensive VR application came with its set of challenges. This chapter dives into the hurdles that the researcher encountered during the development process, highlighting the complexities faced and the solutions explored.

9.2 Unity's Learning Curve

Unity, while a very powerful tool, posed a steep learning curve. The vast array of features and tools demanded extensive research and experimentation, slowing down the development process tremendously. Overcoming this challenge required the researcher to persistently adapt and experiment with different ideas and approaches. This led to a much faster pace of development in the later stages, as the researcher's knowledge of the Unity engine expanded beyond expectations.

9.3 Cybersickness and Delayed Development

Cybersickness, a common challenge in VR, proved to be a stumbling block for the researcher to carry out testing in early development. While only being able to effectively test the application by being immersed in the virtual environment, led to the researcher experiencing extensive cybersickness symptoms, such as nausea, disorientation, and headaches. This significantly reduced development time in the early stages. However, with repeated exposure and perseverance, the researcher gradually adapted, and the symptoms subsided, allowing for smoother, 'pain-free' testing in the later stages.

9.4 Socket Interactors and Collider Challenges

Implementing socket interactors and dealing with collider issues proved to be one of the most intricate challenges. Objects' colliders intercepting with socket interactor colliders led to unexpected and erratic behaviour of the components, demanding meticulous debugging and problem-solving. The absence of online resources in this regard meant resorting to diving deep into problem-solving strategies and continuous, back-to-back testing.

Colliders, a crucial aspect for object interactions, unexpectedly impacted component functionality far beyond expectations. Objects' colliders did not always behave as expected, leading to unforeseen consequences in user interactions. The researcher spent numerous weeks debugging these issues.

9.5 Limitations and Future implementations

Considering the future, the vision of integrating additional tools, like a screwdriver for screw interactions, highlighted the complexity of interactive elements. While valuable, the intricate mechanics and animations require careful planning and scripting, therefore a longer time frame than what was available at the time, was needed.

Similarly, case cables, a fundamental aspect of hardware assembly, presented challenges due to its fine motor skill requirements, necessitating an entirely different approach. This approach will need a much longer time frame for implementation and is, at this stage, considered as future work.

An additional avenue for exploration is also recommended for future development. Integration of peripherals and diverse components with varying specifications lies high on the list of additional features. However, the time limit and scope of the project did not allow for the implementation of such advanced features. Exploring these possibilities will enrich the user experience, and the application will act as an advanced educational tool, providing insights into the complexities of computer hardware compatibilities.

9.6 Summary

In summary, the development process of the VR application posed significant challenges, including Unity's complexity, cybersickness-induced delays, and intricate issues with socket interactors and colliders.

While certain features were deferred for future work, these challenges have provided valuable learning experiences and opportunities for refinement. The chapter underlines the resilience needed in VR development, shaping the path for iterative improvements and future advancements in the application.

Chapter 10: Conclusion

Coming to an end, this project, and its extensive journey in the world of virtual reality hardware assembly represented a merging of innovation, creativity, and technical expertise. Through meticulous design and development, this immersive VR experience now stands as testament to the fusion of education and technology. By immersing users in a hands-on environment where they can intricately assemble computer components, this project fosters not just knowledge, but a profound understanding of the intricate process.

From tackling the challenges to envisioning future expansions, this project has been a transformative endeavour. It marks the peak of countless hours and a step forward into the future of education, where virtual reality becomes a power tool for experimental learning.

As this project concludes, it leaves behind a trail of innovation, promising a vibrant future where immersive technology reshapes how we learn complex subjects. In the end, this project is more than just a digital creation.

References

Amazon. (2023). *Oculus Quest All-in-one VR Gaming Headset*. Retrieved from Amazon: https://www.amazon.com/Oculus-Quest-All-Gaming-Headset-android/dp/B07PRDGYTW

Ambrosio, A. P., & Fidalgo, I. R. (2020). Past, present and future of Virtual Reality: Analysis of its technological variables and definitions. *Culture & History Digital Journal*, 2-3.

Baniasadi, T., Ayyoubzadeh, S. M., & Mohammadzadeh, N. (2020). Challenges and Practical Considerations in Applying Virtual Reality in Medical Education and Treatment. *Oman Medical Journal*.

BlenderKit. (n.d.). *Get BlenderKit*. Retrieved from Blender Models: https://www.blenderkit.com/asset-gallery?query=category_subtree:model%20order:-created

Brush, K., & Silverthorne, V. (2022, November). *Agile Software Development*. Retrieved from TechTarget: https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development

Caserman, P., Garcia-Agundez, A., & Gamez Zerban, A. (2021). Cybersickness in current-generation virtual reality head-mounted displays: systematic review and outlook. *Springer*, 1.

Hamad, A., & Jia, B. (2022). How Virtual Reality Technology Has Changed Our Lives: An Overview of the Current and Potential Applications and Limitations. *Int J Environ Res Public Health*.

Hussain, A., Shakeel, H., Hussain, F., Uddin, N., & Ghouri, T. L. (2020). Unity Game Development Engine: A Technical Survey. *University of Sindh Journal of Information and Communication Technology*.

JDTech Solutions. (2020). *PC Virtual Lab*. Retrieved from PC Virtual Lab: http://pcvirtuallab.com/

Kromenaker, C. (2013, March). *When to Inherit from MonoBehaviour*. Retrieved from Clark Kromenaker: http://clarkkromenaker.com/post/unity-monobehaviourusage/#:~:text=When%20you%20create%20a%20new,you%20don't%20have%20to.

Kunhar Games. (2021). *Our Games*. Retrieved from Kunhar Games: https://kunhargames.com/games-2/

LaRocco, M. (2020). Developing the 'best practices' of virtual reality design: industry standards at the frontier of emerging media. *Journal of Visual Culture*.

Leung, T., Zulkernine, F., & Isah, H. (2018). The use of Virtual Reality in Enhancing Interdisciplinary Research and Education. *ResearchGate*, 1-2.

Lowood, H. E. (2023). *Virtual Reality*. Retrieved from Brtiannica: https://www.britannica.com/technology/virtual-reality

Nancarrow, C., & Brace, I. (2000). Saying the "right thing": Coping with social desirabiltiy bias in marketing research. *Bristol Business School Teaching and Research Review*, 3.

Nesbitt, K., & Nalivaiko, E. (2018). Cybersickness. *Springer*.

Nielsen, J. (2000). Why you only need to test with 5 users. *AlertBox*, http://www.useit.com/alertbox/20000319.html.

Nurnberger, M., Klinger, C., Witte, O. W., & Brodoehl, S. (2021). Mismatch of Visual-Vestibular Information in Virtual Reality: Is Motion Sickness Part of the Brains Attempt to Reduce the Prediction Error? *Frontiers in Human Neuroscience*, 1.

Pahuja, S. (2015, February 15). *Scrum for Individuals*. Retrieved from InfoQ: https://www.infoq.com/news/2015/02/personal-scrum/

Pirker, J., Dengel, A., Holly, M., & Safikhani, S. (2020). Virtual Reality in Computer Science Education: A Systematic Review. *ACM*.

Rebelo, F., Noriega, P., Duarte, E., & Soares, M. (2012). Using Virtual Reality to Assess User Experience. *Human Factors The Journal of the Human Factors and Ergonomics Society*.

Saleem, H. (2023, January 30). *10 Best Tools for VR Development of 2023*. Retrieved from Talentverse: https://www.talentverse.co/blog/best-tools-for-vr-development#:~:text=1.,console%2C%20mobile%2C%20and%20web.

Stengel, R. F., & Furman, B. (1996). The history of virtual reality: An interactive timeline. *IEEE Virtual Reality Annual International Symposium*.

The Irregular Corporation. (2018). *PC Building Simulator 2*. Retrieved from PC Building Simulator: https://www.pcbuildingsim.com/

Turner, N. (2016). *Capturing user feedback with Microsoft's product reaction cards*. Retrieved from UX for the Masses: https://www.uxforthemasses.com/product-reaction-cards/

Unity. (2023). *What is the Unity Asset Store and how do I purchase Assets?* Retrieved from Asset Store: https://support.unity.com/hc/en-us/articles/210142503-What-is-the-Unity-Asset-Store-and-how-do-I-purchase-Assets-#:~:text=The%20Unity%20Asset%20Store%20is%20home%20to%20a%20growing%20library,examples%2C%20tutorials%20and%20Extension%20Assets.

Appendix A: Ethical Clearance Certificate

UNIVERSITY OF THE FREE STATE
UNIVERSITEIT VAN DIE VRYSTAAT
YUNIVESITHI YA FREISTATA
UFS UV

GENERAL/HUMAN RESEARCH ETHICS COMMITTEE (GHREC)

11-May-2023

Dear Prof Lizette De Wet

Continuation/Report Approved

Research Project Title:
CSI Honours Projects Ethical Clearance

Ethical Clearance number:
UFS-HSD2020/0524/1505/21/22/3

We are pleased to inform you that the application to extend your ethical clearance has been approved. Your ethical clearance is valid for twelve (12) months from the date of issue. We request that any changes that may take place during the course of your study/research project be submitted to the ethics office to ensure ethical transparency. furthermore, you are requested to submit the final report of your study/research project to the ethics office. Should you require more time to complete this research, please apply for an extension. Thank you for submitting your proposal for ethical clearance; we wish you the best of luck and success with your research.

Yours sincerely

Dr Adri Du Plessis
Chairperson: General/Human Research Ethics Committee

Adri Du Plessis
Digitally signed by Adri Du Plessis
Date: 2023.05.18 20:06:39 +02'00'

205 Nelson Mandela Drive
Park West
Bloemfontein 9301
South Africa

P.O. Box 339
Bloemfontein 9300
Tel: +27 (0)51 401 9337
duplessisA@ufs.ac.za
www.ufs.ac.za