UNIVERSITY OF TARTU
Institute of Computer Science
Software Engineering Curriculum

Masud Rana

Open and Accessible Education with Virtual Reality

Master's Thesis (30 ECTS)

Supervisor(s): Ulrich Norbisrath, Ph.D.

Tartu 2022

Open and Accessible Education with Virtual Reality

Abstract:

Virtual Reality (VR) is on the rise again. It is currently unfolding its true potential with the availability of head-mounted and standalone VR headsets. Standalone VR headsets are becoming more available and affordable, making it an excellent prospect to use them as a regular educational device. However, to use a standalone VR headset, especially the here targeted Meta Quest 2, as a standard educational device is still challenging. The official Meta store does not contain most of the popular 2D apps, and it does not have Google Play or Google Play services installed either; most of the 3D apps are either games or developed with a focus on very specific tasks like 3D design. Additionally, it is hard to switch between two 3D apps as they cannot be used in parallel. There is no default and simple affordable way of using and visualizing external keyboards. Finally, good guides for using the Quest 2 headset for day-to-day educational computing, are rare and incomplete. This makes it much more difficult for students to maximize their learning efficiency in such an environment. This thesis will illustrate the research, development, and preparation for using the targeted Meta Quest 2 VR headset for day-to-day educational purposes. It will summarize the observations and implementation experiments into a practical guide to make such a virtual reality environment more accessible.

Keywords:

Virtual Reality, Learning Environment, Open Source, Meta Quest, Godot, Termux, Elixir

CERCS: P170 Computer science, numerical analysis, systems control

Avatud ja juurdepääsetav haridus virtuaalreaalsusega

Lühikokkuvõte:

Virtuaal Reaalsus (VR) on taas tõusul. Seadete laialdane kättesaadavus, nii eraldiseisval kui ka peakomplekti kujul, on tohutult kasvatanud VR potentsiaali. Eraldiseisvaid VR seadmeid on leida järjest lihtsam, ja soetada järjest odavam - see avab suurepärase integratsioonide pärusmaa haridus sektoris. Kuigi eraldiseisva VR seadme, nimelt siin käsitletud Meta Quest 2, rakendamine hariduslikel eesmärkidel pole ilma katsumusteta.Ametlikku Meta poodi ei ole kaasatud kõige populaarsemaid 2D programme ning puudub ka tugi Google Play teenusele. Enamik 3D rakendusi kuuluvad kas mängude hulka, või on arendatud eesmärgiga lahendada väga suunatud ülesanne, näiteks 3D disainimine. Lisaks ei ole toetatud rakenduste taustale viimine, ehk pole võimalik lülitada end ümber kahe erineva 3D rakenduse vahel, et neid paralleelselt kasutada. Puudub vaikimisi lihtne viis välise klaviatuuri liidestamiseks. Raske on ka leida terviklikke juhiseid seadete rakendamiseks päevast-päeva kestvas õpikeskkonnas. Näiteks Quest 2 seadme kasutamine arvutuslike probleemide lahkamiseks. See kõik teeb õppe jaoks VR kohandamise keeruliseks nii õpetajale kui ka õpilasele.Lõputöö illustreerib uuringuid, arengusamme ning tegevusplaane Meta Quest 2 seadme kohandamiseks igapäevase õppe tarbeks. Tuues esile tähtsad märkused ja protsessid, et teha VR õpikeskkondade püstitamist kergemini kättesaadavaks.

Võtmesõnad:

Virtuaal Reaalsus, Õpikeskkond, Avatud lähtekoodiga, Meta Quest, Godot, Termux, Elixir

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaat-juhtimisteooria)

Acknowledgments

I want to convey my gratitude to my supervisor, Dr. Ulrich Norbisrath, who supported and directed me on the right track.

I again share my acknowledgment with my colleague, Meelik Kiik, who helped me with the Estonian translation. I also thank Ashfaq H Ahmed, Vishal Tirkey, and Shrikant Shivaji Pawade for providing valuable feedback.

Table of Contents

1 Introduction

Creating an ideal learning environment is challenging and involves a lot of contradictory factors. The term ideal learning environment can differ from the perspective of someone's age, physical ability, preference, or financial ability.

An ideal learning environment is a set of tools and opportunities that push toward successful learning and balances individual learners' need to meet their unique objectives. Moreover, it is a place or space where other learners can support or influence one's learning experiences. Finally, this place needs to be satisfying for learners and fulfill their needs. [1]

Virtual Reality (VR) is a simulated environment that can mimic the actual physical or completely different backgrounds[1]. Although, until now, the primary consumers of Virtual Reality are for entertainment, it has also started to be used for educational purposes [2]. Furthermore, for educational applications, this technology has a vast perspective.

As VR can simulate the real world and incorporate virtual objects, it is quite possible to make a decent virtual educational environment that would be costly or not even possible in some cases due to various real-life constraints. This virtual simulated learning environment can solve many challenges of current online learning systems.

In this thesis, I have tried existing tools like Termux to run Linux, VS Code Server, and Godot to experiment with passthrough features to make our target Meta Quest 2 VR environment learning-friendly. A web-based tool named Buddi is a work in progress to solve the challenges of storing, sharing learned materials & notes, and collaborating inside the VR environment while consuming learning materials through the browser. Buddi is a 2D web application that intends to solve general learning challenges in VR in more straightforward and usable ways. This utility tool will have features like notetaking, file sharing, or note sharing and will not have features like lecture playback or evaluating a student. So, it is not envisioned as a replacement for learning and examining platforms like Moodle. Besides this, it also reduces distraction and provides more focus while consuming some extra power of VR. Finally, a VR learning environment guide will be provided to solve students' daily tasks based on related works, experiments, and requirements of the targeted student persona.

1.1 Motivation

Virtual reality can be an excellent solution for many educational challenges. Currently, VR headsets are quickly becoming more affordable and powerful. For example, our target device Meta Quest 2, can be purchased at a price under USD 400. This device is powerful enough to fulfill computation requirements for most learning tasks compared to an average Chromebook within this price range. Furthermore, it provides an immersive experience that enables working/studying in a decent environment irrespective of the user's location. Another critical advantage that VR delivers is the ability to multitask with multiple virtual screens in a small and constrained space, compared to using numerous physical screens on a large desk with a computer. This device can be handy for students with financial disabilities or who cannot afford a decent computer if they can do similar stuff on this device.

One of the critical problems we have experienced in education during the pandemic is the restriction on physical collaboration regarding co-location and spatial togetherness. VR is

[1] https://en.wikipedia.org/wiki/Virtual_reality

the closest digital environment compared to the physical environment. Therefore, it can solve most of the issues faced in education due to the pandemic restriction. Furthermore, let us consider a student who is physically disabled and cannot attend a physical school. He can use the power of VR to attend classes and labs and collaborate almost similarly to physical classes. Therefore, VR headsets can be helpful for students if current challenges can be solved in a more usable way.

1.2 Requirements

While gathering requirements for this thesis, I used the concept of concrete personas and respective user stories/scenarios to extract the requirements according to the concrete input. I have considered three types of user personas, and the below subs section will explain them, and corresponding user story and requirements extraction will be done.

1.2.1 Disadvantaged student

The user persona of a disadvantaged student represents a financially needy student who cannot afford a computer but can afford a mid-level android phone or can get a device like Chromebook from school.

NAME

Robin Uthappa

TYPE

Disadvantaged

Goals

- Study hard and learn advanced technology
- Get a top place in the school
- Attend online classes
- Build a strong study network

Demographic

♂Male 18 years

Student

Quote

" If I had the opportunity to access a modern computer, then I could do my regular study stuff more efficiently and I could explore new things online. Also, I could learn programming with it. "

Technologies

Background

- Low to mid-level income family
- Lives in a small city that has internet availability
- Can afford a mid-level Android device or can get a Chromebook for free from the school

Browsers

Motivations

- Self-directed learning
- Self-belief
- Adaptive to new technology
- Hard-working
- Disciplined while learning study materials

Frustrations

- Didn't have access to a modern computer
- Couldn't be able to use the power of the internet to learn additional material properly
- Printing additional study materials are costly
- Couldn't be able to collaborate with other students online in the group project
- Mobile screens are small to do assignments or do programming
- Working with one screen while learning and applying different sources is cumbersome.

Figure 1: Disadvantaged student persona[2]

[2] https://pixabay.com/photos/portrait-people-adult-africa-3157821/

Example Scenarios:

Title: Robin wants to learn additional information regarding black holes

In physics class, Robin's teacher touched on the topic of black holes. Robin was fascinated. He remembers from the class that black holes are gigantic, and light cannot pass from them due to substantial gravitational force. His curiosity triggers him to know more about it, and he wants to answer the questions like how black holes was formed in the first place, and if light cannot pass from black holes, how can it be recognized? Most importantly, Robin thinks that if he can see some video documentary or detailed document about black holes, then he might get a better idea about it. Therefore, Robin started to use the internet on his smartphone with a 5inch display to get more information about black holes. However, he ended up frustrated with the small screen size, with lots of different information on the internet, and he could not unify them because he could not see them side by side. He cannot print them as printing additional materials is beyond his financial capability.

Title: Robin has a group assignment to submit

Robin's chemistry teacher gives the class a group assignment where each group consists of four students, and the task is to write the properties of atoms from the periodic table. Each group needs to submit properties of at least 20 atoms. So, the group members decided to divide the task individually, where each member had to complete the properties of 5 distinct atoms from the periodic table. All the group members have access to a personal computer except Robin so that other members can work jointly with a shared document through the internet. As Robin did not have an internet-enabled device with a big screen, he had to collaborate with other students by physically meeting them or via phone. However, collaborating through phone calls is a financial burden for him, and a physical meeting is not always possible due to the group members' availability or the short assignment deadline.

Extracted Requirements:

From the abovementioned persona and scenario, the following requirements for the proposed system are extracted:

RQ1: It is possible to stream videos from sites like YouTube and others so that Robin can watch lectures and tutorials about black holes.

RQ2: Reading digital materials like pdf or doc is possible so that Robin can download additional material in digital form and read them when he wants to.

RQ3: It is possible to take digital notes while consuming learning materials so that Robin can compare different information about black holes.

RQ4: Writing and editing computer programs so that it is possible to learn basic programming as Robin is interested to learn to program.

RQ5: Writing assignment reports digitally with basic text formatting so that Robin can easily make corrections and collaboration while contributing to group assignments.

RQ6: Sharing files with other students so that Robin can share his work with the group members in a group assignment.

RQ7: Collaborating in a group assignment

1.2.2 Restricted Student

The user persona of a restricted type of student represents a student restricted from doing physical collaboration due to a pandemic, physical disability, or other reasons.

NAME

Melbin Jose

TYPE

Restricted

Goals

- Study hard and learn advanced technology
- Get a top place in the school
- Attend online classes
- Build a strong study network
- Practice theoretical learning from lectures in labs

Demographic

Male 19 years

Student

Quote

" Pandemic makes study life worse, and I can not be able to attend physical classes and cannot perform group works physically. It would be better if I could collaborate with other students online more realistically and efficiently "

Technologies

Background

- Cannot attend classes physically due to pandemic restriction
- Mid-level income family
- Lives in a moderate city that has internet availability
- Can afford a decent modern computer

Browsers

Motivations

- Self-directed learning
- Self-belief
- Adaptive to new technology

Frustrations

- Traditional online classes are less collaborative
- It is tough to keep concentration while attending class from home
- It is frustrating to participate in theoretical classes and demonstrations without actually practicing it physically or close to physical means.

Figure 2: Restricted student persona[3]

[3] https://pixabay.com/photos/human-interest-human-potrait-nature-5494916/

Example Scenarios:

Title: Melbin participates in online classes

Melbin comes from a decent mid-income family with a proper personal laptop and internet access. Things had been going well for him until the covid-19 pandemic hit, and he had to attend classes online and was restricted from meeting friends. After attending online classes for some months, he started to feel lonely, and online courses seem not that lively. For group work, the online platform uses breakouts that feel boring due to the synchronization issues among members, and sometimes it is hard to follow others. He also started to miss lab classes where he could discuss with other students and instructors and do experiments. As he cannot control pandemic situations, he expected online learning to be livelier and closer to physical education.

Extracted Requirements:

From the abovementioned persona and scenario, the following requirements for the proposed system are extracted:

RQ8: The system should enable the opportunity to attend online classes more lively manner

RQ9: Collaboration with students and teachers in the lab work should be possible

1.2.3 Advanced Student

The user persona of an advanced type of student represents an advanced student who wants to use the latest technology to learn the educational content more efficiently.

NAME

Pradip Kumar

TYPE

Advanced

Goals

- Study smart and use advanced technology
- Get a top place in the school
- Learn by experiencing
- Build a strong study network
- Experiment with advanced technology to do cool things

Quote

" It is better to learn and practice smartly with technology than just use pure imagination."

Demographic

♂Male 19 years

Student

Background

- Mid-level income family
- Lives in an advanced city that has good internet availability
- Can afford a decent modern computer

Technologies

Motivations

- Self-directed learning
- Self-belief
- Adaptive to new technology

Frustrations

- Exploration of the world with 2D maps and screens doesn't feel so engaging.
- Multiple PC screens are smart ways to learn and practice simultaneously but they are not portable.

Browsers

Figure 3: Advanced student persona[4]

[4] https://pixabay.com/photos/man-male-model-indian-denim-pose-7040262/

Example Scenarios:

Title: Pradip wants to learn the history of Ancient Egypt

Pradip is an intelligent kid who constantly thinks about doing something brilliant rather than just doing it the traditional way. In the history class, he gets to know some history of Ancient Egypt, and that triggers his curiosity to learn more about it. So, he searched the internet for documentaries about ancient Egypt history and watched some of them. However, he felt that if he could be part of that history virtually, then make much more impact on his experience-based learning. Another issue he felt while making notes and watching the documentary was that he had to switch between windows on his laptop; multitasking does not seem convenient in this approach.

Extracted Requirements:

From the abovementioned persona and scenario, the following requirements for the proposed system are extracted:

RQ10: Visiting different places online to learn geography and history by experiencing them.

RQ11: Multitasking with multiple screens to learn and apply other study materials

1.2.4 Extracted Set of Requirements

The purpose of this requirement gathering based on the upper mentioned three personas is to find out tasks that need to be performed in an educational environment. The combined list of extracted requirements is as follows:

RQ1: It is possible to stream videos from sites like YouTube and others so that it is possible to watch lectures and tutorials.

RQ2: Reading digital materials like pdf or doc is possible.

RQ3: It is possible to take digital notes while consuming learning materials.

RQ4: Writing and editing computer programs so that it is possible to learn basic programming.

RQ5: Writing assignment reports digitally with basic text formatting so that easy to make corrections and collaboration.

RQ6: Sharing files with other students

RQ7: Collaborating in a group assignment

RQ8: The system should enable the opportunity to attend online classes more lively manner

RQ9: Collaboration with students and teachers in the online lab work should be possible

RQ10: Visiting different places online to learn geography and history by experiencing them.

RQ11: Multitasking with multiple screens to learn and apply other study materials

2 Related Work

2.1 Immersed

Immersed is a VR application for collaboration and personalization of virtual workspace developed by Immersed Inc. Its main goal is to increase productivity[5].

Users must install the Immersed app from the meta store to use this application. After that, they will get a pairing code and username, then need to enter those to the immersed official site to download and install Immersed desktop app to the computer. Then it is possible to connect from the desktop to Immersed VR app with the pairing code and username. User will immediately get their desktop screens in Immersed, and some tutorials on using will be played out to get started with the Immersed application.

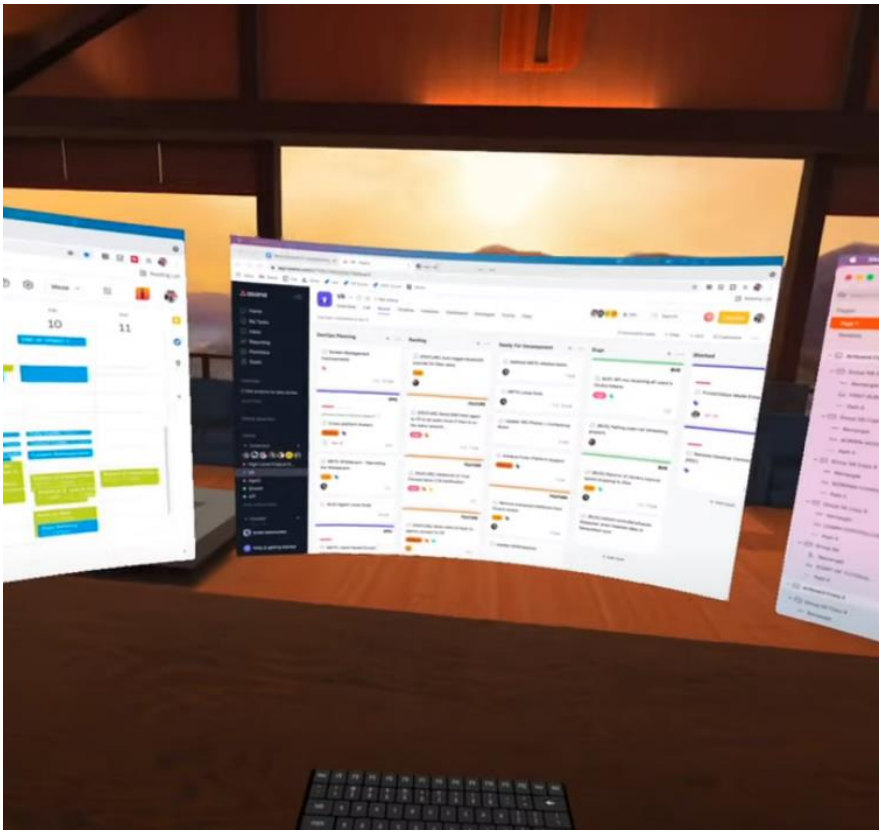

Figure 4: Immersed with multiple screens from desktop[6]

With Immersed, it is possible to use five virtual desktops simultaneously along with collaboration opportunities. Immersed offers multiple screen sharing, telepresence, and remote whiteboarding during collaboration. Immersed has introduced external keyboard tracking for oculus-supported keyboards. For other keyboards, tracking is not available, but with calibration of keys, it is possible to bring their layout with passthrough on the specified keyboard area to the VR space. Another way to make keyboard typing easier is to use the passthrough only for the keyboard area. Furthermore, with Immersed, it is possible to bring personal phones into the workspace. So, it is possible to get phone messages and calls from Immersed, although only iPhone and iPad are currently supported.

[5] https://immersed.com/

[6] https://www.youtube.com/watch?v=1TLC2tdImZ0

Another appreciable feature of Immersed is its remote whiteboarding feature. It allows people to collaborate remotely through a virtual whiteboard in a livelier way.

While testing Immersed with Ubuntu desktop version 20.04 and Meta Quest 2, some lagging was observed due to Nvidia graphics driver issues in the system. Later, I tried again with Windows 10, and it worked without noticeable lag. I had experienced my two physical screens and one virtual screen provided by Immersed. Virtual screens worked similarly to physical screens, and I believe they improved my productivity. However, due to the low resolution, I had to make the screens much larger.

Immersed can fulfill all the requirements mentioned in section 1.2 except RQ9 and RQ10. Furthermore, Remote whiteboarding can be an excellent option for collaboration and brainstorming in a group assignment (RQ7). It will be good to have keyboard tracking and keyboard passthrough features in the proposed educational environment. I am adding this requirement to the set of requirements mentioned in section 1.2 as below:

RQ12: The system should render the physical keyboard as a virtual object in the VR environment with keys overlay and passthrough in the keyboard area so it will be easy to type.

The significant difference between Immersed and this thesis is the need for another PC to use. Immersed uses the computer's power for processing, while this thesis primarily focuses on standalone VR solutions for day-to-day educational computation needs.

2.2 Meta Horizon Workrooms

"Meta Horizon Workrooms" is a similar application as Immersed mentioned in 2.1, developed by the company Meta [3]. It is designed to focus on remote work and collaboration in mind. Horizon Workrooms allow users to work closely in a virtual environment, and they can communicate, present or collaborate.

Figure 5: Meta Horizon Workrooms [3]

Meta Horizon Workrooms is now in beta version and available for free in Meta Quest 2 store. To get started with the Horizon Workrooms, one of the team members needs to create a Workrooms team on https://www.oculus.com/workrooms/. If a team is already

made, team members can send a request to join to other members. After the account creation is completed, the Horizon Workrooms app needs to be installed from the Oculus store and follow instructions to pair the headset to the user account.

Meta Horizon Workrooms does not support multiple virtual screens like Immersed and does not support bringing the personal phone into virtual space yet. The lack of support for virtual screens makes Meta Horizon Workrooms different from Immersed as it focuses on collaboration and remote work, whereas Immersed focuses on productivity. It also has features like a collaborative virtual whiteboard.

Meta Horizon Workrooms is available free to test out with real users. Currently, Meta Horizon Workroom relies on 3D avatars to represent the users with spatial audio. It is an excellent approach to bring some interactivity during collaboration in remote work. However, we may see a more improved version of avatars for this application. It might be possible to generate lifelike avatars and capture natural expressions [4].

So, Meta horizon workrooms are suitable for collaboration and day-to-day computation in VR, but like Immersed, it also needs a computer to do the processing. However, features like virtual whiteboarding, keyboard passthrough, virtual presentation, or changing seating arrangement are pretty good to have in a learning environment during collaboration. It would be good to have all these features that consume the processing power of a standalone VR headset, e.g., Meta Quest 2. As in this thesis, I am trying to use the standalone VR headset to solve the computation required for education.

2.3 Project-Based 3D Virtual Learning Environment for English

Xiaoluo et al., 2022 [5] proposed a 3D project-based learning environment to facilitate practical English learning and teaching. The idea was to achieve experiential teaching modes and provide interactivity to students. They have developed the proposed environment with the help of a Virtual World named Second Life (TM) or SL, where the user or residents of SL can build, add, and modify its content. SL provides powerful creation tools, including 3D modeling tools and interactivity that can be achieved with Linden Script Language (LSL) scripting language. SL supports worldwide navigation by walking, teleporting, flying, or running. Furthermore, it is also possible to create new interactive objects, change avatar appearances and surroundings, and the possibility to communicate through spatial voice [6] with other users[7].

In this paper, they have done a case study on "English Community," a virtual learning environment in SL built by Three Immersions, Inc. for Aijia Cambridge English Training Center. This project was created for students of age 10-20 who know basic English grammar, words, and phrases but with lower communication skills. It provides a 3D virtual learning environment in Second Life (TM), and students are recommended to talk in English and might also find some common interests in talking.

[7] https://secondlife.com/

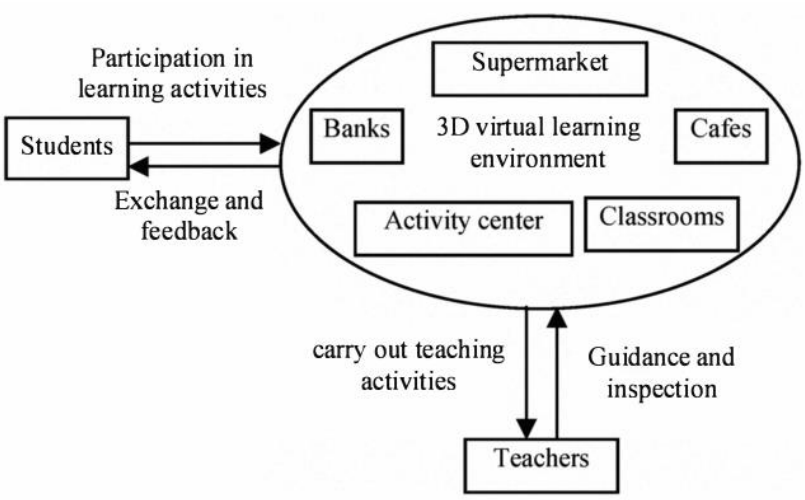

Figure 6: The structure of the 3D virtual environment for "English Community" [5]

"English Community" in Second Life (TM) was achieved with 3D modeling to mimic actual worlds and some futuristic structures to keep students' attention. Scripts are also added to 3D objects to achieve something like playing PPT, billing, or selling goods.

Figure 7: The 3D virtual environment of "English Community" [5]

In this virtual learning environment, teachers can set up tasks for the students, "go to the supermarket," and students can also add their own choice of activities.

For example, the teacher allocates an individual or a group with a shopping task. All the goods on the shelves have spelling, prices, and pictures in the supermarket. The teacher will answer students' questions or problems through broadcasts or group 1M. When the student or student group gets the specified category they were looking for, they click on the product and need to answer some questions. If the answer is correct, then they get

points. If they have difficulties answering the question, it is possible to ask for help from a teacher or other students.

Finally, students will line up for the checking-out process, and they can discuss today's shopping experience. Furthermore, they have to report the price of each item; the teacher will confirm and ask some more questions, and they get credits for the correct answer. After shopping is over, students have to submit a feedback survey about how this course goes.

Xiaoluo et al., 2022 [5] mentioned two critical challenges during this study. The first one is the skills issue of students using Second Life. So, before using the environment, they must complete a series of Second Life tutorials on walking, communicating, or teleporting [7]. Secondly, teachers have to be technically sound to solve technical difficulties in the virtual world and communicate properly. For example, when all students talk together [8].

SL is a good environment where it is possible to create new objects and add interactivity with scripts. However, it is unsuitable for a standard educational environment to fulfill day-to-day computation needs. The "English Community" is an excellent example of its application to solve specific academic challenges. However, this thesis aims to create or provide guidance for creating a standard computation environment in VR that can fulfill computational needs in education. So, SL might not be a good fit to meet our objectives.

However, the challenges that the author of this paper mentioned about the learning curve requirements before using the "English community" should be eliminated in our proposed learning environment. The proposed environment should be as intuitive as popular computation Operating Systems like Linux, Android, or Windows, requiring less learning curve. I am adding to the requirements with the requirements list mentioned in section 1.2 as below:

RQ13: The system's operating system should be intuitive as popular operating systems like Android, Windows, Linux, or our targeted Meta Quest 2 environment that I have experienced during my experiments so that the learning curve will be less.

2.4 Programmable Virtual Reality Environments

Nanlin et al., 2021 [9] presented a virtual reality environment where it is possible to manipulate or create surrounding objects with programming. The programming is done by custom visual block language, and this project is made to help the students learn programming and spatial thinking efficiently.

The presented solution has features of writing, executing, pausing, saving, and loading code. It also provides some guidance about the system and general knowledge of programming. The paper provides the system's technical and design elements and first impressions.

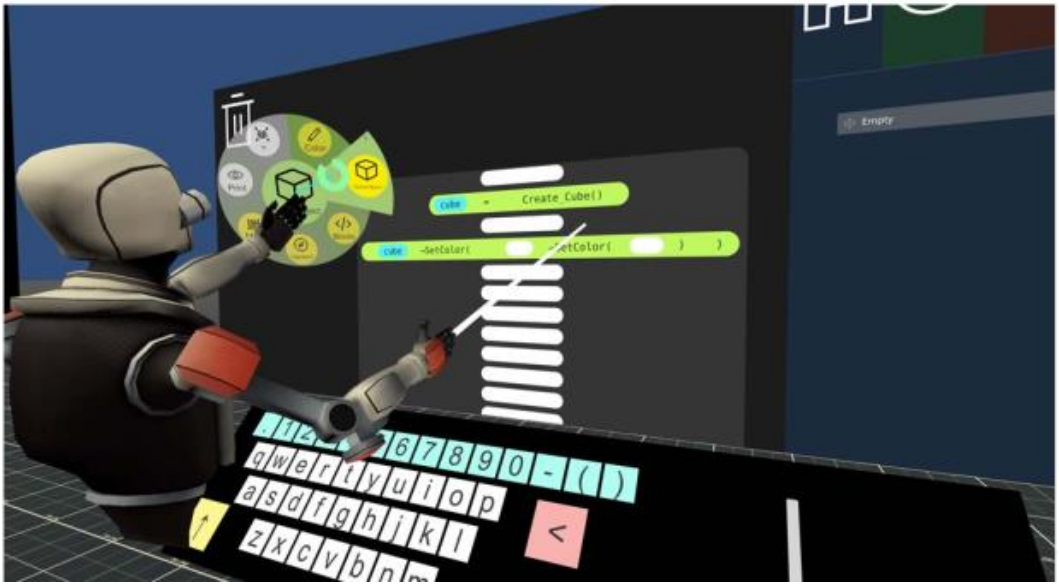

Figure 8: The programmable virtual environment [9]

The proposed virtual environment includes a coding panel, menus for managing and editing codes, and a two-dimensional code editor. With the code, it is possible to instantiate and manipulate surrounding objects and stop and pause code execution. The environment is made with the help of the Unity engine[8].

This application is built with three-layer architecture as follows:

- Presentation layer: It is responsible for the user interaction and input techniques that allow interaction with the programmable VR environment. This layer emits signals with the user's interaction and tells the logic layer to generate, execute or compile the code.
- Logical layer: Logical layer is responsible for all execution, compilation, and generation of code in the run time. Code is executed by interpreting Lua code and calling corresponding Unity C# runtime bindings.
- Data layer: Data layer is responsible for memory management.

They have used MonoSharp and Lua interpreter written in C# for the .NET Mono for Unity. They used a six-degree of freedom (6DOF) controller for the interaction design [Figure 9].

[8] https://unity.com/

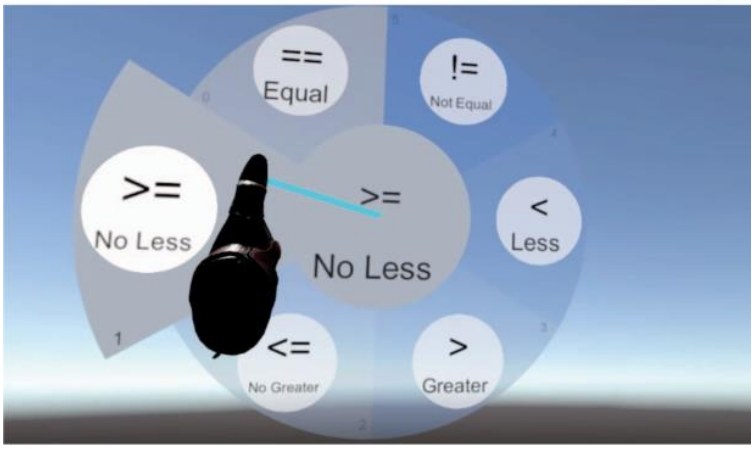

Figure 9: Uses of 6DOF controller to select "No Less" [9]

Nanlin et al., 2021 [9] mentioned this work as a preliminary work, and they have faced some performance issues with Lua, and in the future, they want to replace that with Roslyn C#. Moreover, they also plan to do formal user studies in the future.

Programmable Virtual Reality Environments is an excellent contribution to visualizing and learning programming with more focus and motivation. The objective of this work is almost like Scratch[9]. *Scratch* is an application built for Kids to improve their computational and creative thinking, problem-solving, collaboration, teaching, and learning in computing[10]. However, like Scratch, Programmable Virtual Reality Environments is not a solution to serious software development. Abstraction, portability, testing, or debugging are not ready to be used for actual software development.

Most importantly, it has not been developed to perform software development [9]. This thesis aims to prepare an environment where real software development work is possible, along with other computational needs for education. Lastly, as 3D VR apps are not easily switchable, it is challenging to use this application parallelly with other applications[11]. New requirements for the proposed system will be introduced in the next section to solve this app switching problem.

Although this application is not built for serious software development, it can be one good example of coding in a Virtual Reality environment. Learning basic programming concepts can be possible with this application to fulfill our requirement, RQ4. Serious Visual programming languages like Node-RED[12] or Blueprint[13] might be a good fit to write and execute similarly in the VR environment.

2.5 Virtual Reality Classroom Applied to Science Education

The study's goal currently being reviewed is to develop a virtual reality classroom. The authors have created a series of virtual reality environments and evaluated their effectiveness in achieving that [10]. This paper illustrates the procedure, execution,

[9] https://scratch.mit.edu/
[10] https://scratch.mit.edu/about
[11] https://www.reddit.com/r/OculusQuest/comments/t5sa40/can_the_internet_browser_of_the_quest_2_run_in/
[12] https://nodered.org/
[13] https://docs.unrealengine.com/5.0/en-US/blueprints-visual-scripting-in-unreal-engine/

analysis, and result of the experiments with chemistry and anatomy virtual learning environments that they developed and tested with students to find out effectiveness.

This study used pretest and posttest quasi-experimental research designs. The study is done with 105 students from Taiwan. Students were divided into three groups of 35, and one was a controlled group, which received the traditional study method. The other two groups are experimental and received a VR learning system. The total experiment is done over four weeks. Before the experiment, the pretest was done, and after the experiment, the posttest was done. Moreover, a one-way analysis of variance (ANOVA) was used for data analysis. They have used the HTC Vive VR system for development.

- **Chemical Reaction Experiment:** The developed chemical reaction lab virtual environment has a complete periodic table that can be manipulated with a hand-held handle and can be observable in close range. Each element in the table also has information about that element for extended learning. Students can interact with each element or perform chemical reactions with the elements, giving an immersive reaction experience. Before the experiment, they provided some voice introduction to the VR environment and immersive learning like the traditional ways[14].

Figure 10: Chemical Reaction Experiment [10]

- **Human Anatomy Experiment:** The developed system has a 3D virtual model of the human anatomy that the controller ray click panel can closely look up. Viewing the specific parts of the anatomy at different angles by rotating is possible, and the possibility to zoom is available there. The panel is divided into nine types: muscle, bones, lymph, nerve, Urology, skin, and nine other systems. Like the chemical lab described above, before starting the lab, some traditional voice presentation is done to introduce the system and basic human anatomy[15].

[14] https://www.youtube.com/watch?v=f00RMsvX8UU
[15] https://www.youtube.com/watch?v=lHiZjyISDkA

Figure 11: Human Anatomy Experiment [10]

After the experiment was finished, a posttest was done on three groups. The experimental group's posttest result was significantly higher than the controlled group.

This contribution solves specific educational challenges like the previous two papers (2.3 & 2.4). It would be great if we could run notetaking or browser-based application on top of it and take notes or do further research on what we have learned. Nevertheless, like other 3D apps, to enable note taking, we have to close this app, open the notetaking app, take notes, and rerun the 3D app, or this app should facilitate note taking. Alternatively, maybe some 2D app overlays can be introduced so that the 2D notetaking app can be run along with the 3D application. Therefore, I am adding the following requirement to the list mentioned in 1.2.

RQ14: The system should support running a 2D app overlay within the 3D application.

The virtual reality classroom was an excellent way to motivate students to learn some complex STEM topics. It would be nice if this application also supports collaboration (RQ7) between students and the possibility of taking notes (RQ3) while learning.

2.6 Virtual Reality for Education and Workforce Training

Daniel, 2017 [11] aims to evaluate the efficacy of virtual reality applications for education and workforce training as head-mounted VR devices are quickly becoming more affordable, like Oculus Rift, Samsung Gear VR, and HTC Vive.

This paper proposes two VR training tools for the training mainly focused on the industrial workforce and safety. The first tool is named as "Industrial Workplace Training Tool." It is focused on training low-skilled workers with the fundamentals for accomplishing tasks. This solution is proposed as it is rather difficult to allocate the whole workspace physically

for training purposes only. The proposed virtual training tool will teach the workers about the primary jobs, wearing protective equipment, and identification and use of the tool.

Figure 12: Industrial Workplace Training Tool [11]

The second proposed tool is named "Tool Use and Safety Training Tool," which allows low-skilled workers to perform various experiments with tools. This tool aims to train workers to identify proper tools and uses of protective equipment and determine the wear and tear of the tool that makes the tool unsafe. This training is effective in the sense that it does not create actual harm to the workers but makes them conscious at the same time of substantial damage during the actual work.

Figure 13: Tool use and safety training tool [11]

A time log of the interaction was used to evaluate the worker's performance in the simulated environment. They have logged the controller input, and head position, interaction with objects. They have also simulated the worker's behavior when external forces instantiate some extra material or co-worker. They have also assessed the accuracy by evaluating the uses of tools that are available for him to accomplish tasks.

The proposed tools need validation, and in future work, they aim to validate them with experts and measure effectiveness by reviewing the actual test subject.

As this tool's objective is also to accomplish specific tasks of training the workforce, I have to conclude the same statement I made on 2.5.

2.7 Futuclass

Futuclass is an application with gamified physics and chemistry VR lessons for 7-9 grade students developed by Futuclass OÜ[16]. They are working with schools to provide STEM education in a gamified manner so that it can be more effective, and students feel more motivated to learn.

Futuclass have predefined lesson plans, the teacher has to select lesson plans from the teacher portal, and then VR lessons can be carried out, and at the end, it is also possible to check students' performance[17] .

I have tested the Futuclass reaction balancing lesson, and it provides different challenges to learn and practice reaction balancing. It also narrates how to balance the reaction initially. Solving the reaction balancing puzzles was an exciting experience.

Figure 14: Futuclass reaction balancing lesson

[16] https://futuclass.com/
[17] https://futuclass.com/

This tool is developed to carry out gamified VR STEM lessons to improve students' motivation for learning. This kind of learning environment can be a great fit to fulfill the requirements of the Advanced Persona, Pradip, who likes to learn things smartly by experiencing or by practicing them. Furthermore, gamification in learning might improve the computational thinking of students. It would be nice if students could play the initial instruction of the lesson when required. Also, the availability of a corresponding lesson in text or video format while solving the puzzles might be helpful to learn a little bit further.

3 Experiments

In this thesis, I have experimented with some technologies to fulfill the requirements mentioned in section 1.2. Some of the experiments failed due to complexity and lack of need as a similar solution is already present in our targeted device Quest 2 default environment. Moreover, some of these experiments added new requirements to our list to do more experiments to solve that problem. This section will describe those experiments and outcomes.

3.1 Godot

Our initial idea was to build an environment to unify all the requirements mentioned in section 1.2. Furthermore, one of the fundamental goals of the thesis is to use open-source technologies only. Godot is free and open source and has an MIT license without royalties[18]. So, we decided to use the Godot engine to develop the environment.

3.1.1 Setting Up Godot for Meta Quest development

Installation of Godot and setting up for quest development was not harsh but faced issues as multiple conflicting documents are still present in the Godot official documentation, and sometimes old versions of the documents come up during a google search. The most mentionable issue I have faced while preparing the development setup in Godot is selecting the right plugin for Quest development. As the VR world is racing and new standards are adopted into it, so does the VR plugin in Godot. At the time of this thesis, Godot has three plugins that support Meat Quest development, including Oculus Mobile Plugin (deprecated), OpenVR, and OpenXR[19]. Although Oculus Mobile Plugin is deprecated, this plugin is built explicitly for Quest development. One excellent toolkit named 'Godot Oculus Quest Toolkit' was created with this plugin that has all basic VR implementations[20]. So, it was not easy to ignore this plugin, as I am concerned about the faster project implementation due to time constraints. The below section illustrates some advantages and limitations of different plugins and the final selection.

- **Oculus Plugin**

Godot oculus plugin is specially made for Quest development, and it is available in the Godot addons store and straightforward installation[21]. This plugin is deprecated and will not receive any future development, including Godot 4.0 support. However, as this is the oldest plugin built only for the oculus quest, some excellent VR development toolkits are built on top of it. Another key advantage of this plugin is the possibility of one-click deployment directly from the Godot editor to the quest. This one-click deployment option makes the development and testing much more manageable. However, one key downside of this plugin is that it does not support passthrough. Passthrough support is vital as with the requirement RQ12; we need to facilitate passthrough while using the physical keyboard. My test project with Oculus Plugin can be found on the GitHub link in the reference section [12].

[18] https://docs.godotengine.org/en/stable/about/introduction.html
[19] https://docs.godotengine.org/en/stable/tutorials/vr/index.html
[20] https://github.com/NeoSpark314/godot_oculus_quest_toolkit
[21] https://github.com/GodotVR/godot_oculus_mobile

- **OpenVR plugin**

OpenVR is an open-source API standard accepted by most headset manufacturers and introduced by Valve Software[22]. OpenVR was presented just before the release of OpenXR, and OpenXR is now the latest standard for VR development. Although the OpenVR plugin still exists, all the focus of the Godot team is on OpenXR. Therefore, finding the correct documentation to solve specific issues was complicated with OpenVR. I have tried this plugin and found out that it does not support one-click deployment to quest, and I had to use the side loader to test the application. I also faced some issues with making the controllers work [13].

- **OpenXR plugin**

OpenXR is the latest and future VR development standard for Godot and all popular game development platforms. It was built by the Khronos Group consortium and adopted by most headset manufacturers[23]. OpenXR is at the core of Godot's upcoming version Godot 4.0. Godot team is emphasizing the improvement and addition of new documentation for the OpenXR plugin, and it has good tutorials made by its main contributor BastiaanOlij. However, it is also missing one-click deployment for Meta Quest 2. After searching for an alternative to the quick deployment and testing, I have found the 'Quest Link' that can run a PC VR game into Meta Quest 2 directly from the development environment, and logging messages in Godot's console was also possible.

Finally, as OpenXR is the latest plugin and provides a passthrough feature required for this thesis, I had decided to use it for development. I also installed Godot Oculus Mobile Plugin alongside it to get the feature of one-click deployment to the Meta Quest 2 and used Godot Oculus Quest Toolkit[24]. Furthermore, they can work side by side without issues [14].

3.1.2 Godot to play videos in virtual reality

One of the requirements of the target educational environment is to facilitate video playback and streaming from the internet, RQ1. Godot provides VideoPlayer Node to play videos in WebM, Ogg Theora, and other formats exposed via the GDNative plugin[25]. Godot does not provide any out-of-the-box solution for video streaming; instead, we have to rely on GDNative implementation that allows running native shared libraries at run time. And, with GDNative, compiling the shared library with the engine is not required[26].

I have tried the VideoPlayer Node to play videos from local resources, and it was successful. However, to fulfill RQ1, I have to stream videos from different sites like YouTube. So, it is apparent that I need a full-fledged browser rather than a mere video player or streamer. Moreover, Meta Quest 2 has a default Oculus browser that can fulfill it. Therefore, we decided to use the default Quest 2 browser as a solution to our RQ1.

3.1.3 Godot to enable passthrough features

Godot has added passthrough features with its OpenXR plugin. With the passthrough feature, it is possible to see the outside world through Meta Quest 2 cameras. The Passthrough API provides support to set transparency to the object so that it is possible to

[22] https://github.com/ValveSoftware/openvr
[23] https://www.khronos.org/openxr/
[24] https://github.com/NeoSpark314/godot_oculus_quest_toolkit
[25] https://docs.godotengine.org/en/stable/classes/class_videoplayer.html
[26] https://docs.godotengine.org/en/stable/tutorials/scripting/gdnative/what_is_gdnative.html

make some parts of the viewing window transparent and keep other parts opaque. Our idea was to use this feature to make the physical keyboard visible, so it is easy to take notes inside the VR environment to comply with the requirement RQ12.

This experiment was successful with OpenXR, and it was possible to enable the passthrough when required.

3.1.4 Results

Godot is an excellent engine for all the primary support for 2D development, and it is still relatively new for 3D development and early age for VR development. It is expected to be a more versatile VR solution with the stable release of Godot 4.0. Furthermore, after going through the experiments mentioned above, it was understandable that accomplishing our requirements would be complicated. Because of the lack of previous experience, Godot's built-in support for some specific solutions and community projects built with Godot for Virtual Reality targeting Meta Quest 2. Furthermore, we can achieve enough to show the proof of concept (POC) by other means.

Moreover, most of the requirements mentioned can be solved with our targeted Meta Quest 2's system environment. Therefore, we decided to check some existing tools that run on android and can solve the requirements. Furthermore, suppose we have some requirements that cannot be fulfilled efficiently with existing tools or have some unification problem. In that case, that problem will be solved with new tools. One interesting fact about Godot editor is that it is available to run on Android. So, it is possible to use Godot as a programming learning and game development tool (RQ4) inside Meta Quest 2.

3.2 Termux

Termux is an open source 'terminal emulator' or a computer program that imitates or emulates a video terminal inside another display architecture for Android to enable a Linux environment on an Android device[27]. Termux has package managers that will allow installing other software, including a complete Linux environment. Termux packages are cross-compiled with Android NDK and need to perform compatibility fixes to be able to run them on Android.

Termux supports CMake like build tools and compilers for Rust, C++, Go, Elixir, and many others. Installing interpreters for Python, JavaScript, and other popular interpreted languages is also possible. With the installation of Python, pythons pip package manager is also available and expands the opportunity to install python-based packages.

The initial release of Termux was in 2015; in 2020, Termux ended support for android devices running on Android versions 5-6 and set the minimum required Android version to 7. With the policy changes of Google Play, the updated Termux application is no longer available on Google Play, and currently, the official download source of Termux is Fdroid. As of 2021, Termux is maintained by volunteer software developers. Currently available Termux repositories are main, x11-repo, and root-repo.

[27] https://en.wikipedia.org/wiki/Termux

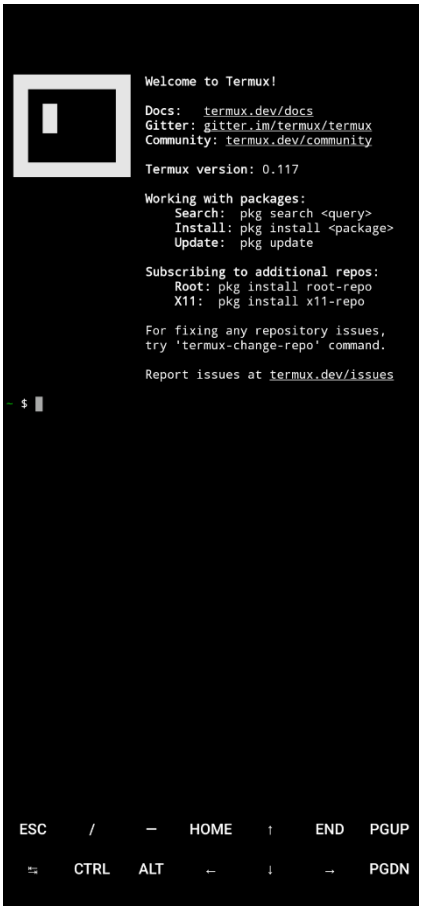

Figure 15: Termux welcome screen

I have tried out Termux to experience its applicability for installing and using different Linux packages and distros. The requirements mentioned in section 1.2 shows that the required educational environment must be able to install educational and utility software. It is essential if we want to run things in the local environment. For example, digital reading materials like pdf or doc require reader software that supports those formats (RQ2). The Linux environment is mature and consists of numerous applications. Therefore, If we can enable Linux features, streaming videos from websites (RQ1), reading digital materials (RQ2), taking notes while consuming digital materials (RQ3), writing and editing computer programs (RQ4), writing assignments and collaboration (RQ5 & RQ7) and sharing files (RQ6) will be possible. As Linux is a popular operating system, and if the user is familiar with it, it might not need much learning curve to use the Meta Quest 2 VR environment (RQ13).

3.2.1 Installing OpenSSH and enabling wake lock

OpenSSH is a utility tool to provide security while communicating with unsecured channels[28]. It is possible to install and use OpenSSH in Termux with the following steps:

- Installing OpenSSH: OpenSSH can be installed with following commands in the Termux shell.

```
pkg upgrade
pkg install openssh
```

[28] https://wiki.termux.com/wiki/Remote_Access

- Set up a password: After installation of OpenSSH, a password needs to be set, and recommended password is 8-10 characters long. While typing the password, it will not be visible.

```
$ passwd
New password:
Retype new password:
New password was successfully set.
```

- Generate RSA public and private key: To authorize SSH connection, a public key and a private key are required. Keys can be generated with the following commands.

```
ssh-keygen -t rsa -b 2048 -f id_rsa
```

After executing this command, id_rsa.pub and id_rsa files will be generated in the same directory. Id_rsa.pub contains a public key, and id_rsa file contains a private key.

- Login to a remote machine: Public key needs to be added to that by the following command:

```
ssh-copy-id -p 8022 -i id_rsa IP_ADDRESS_OF_THE_REMOTE_MACHINE
```

Now to login into that machine, the following commands need to be used:

ssh -p '8022' IP_ADDRESS_OF_THE_REMOTE_MACHINE

Wakelock is a feature that causes the CPU to not go into sleep mode[29]. It might be essential while running some software for a long time, scripts, or a server. To enable Wakelock in Termux, 'termux-wake-lock' and to disable it, 'termux-wake-unlock' can be used[30].

3.2.2 Installation of git

Git is an essential tool for version management of the computer program or software development. Termux main package manager has a git package, and in the default Termux CLI environment running the following command installed git.

```
apt update && apt upgrade
apt install
```

3.2.3 Installation of Codiad on Termux

Codiad is a web-based ide with a simple UI and small footprint. As Codiad is a server-based IDE, I have tested its installation in Termux to check the experience. To install Codiad on Termux, one needs to take the following steps:

- Install wget by running the following command:

```
pkg install wget -y
```

[29] https://www.reddit.com/r/termux/comments/pq1rla/what_is_acquire_wakelock_and_why_is_it_useful_ive/
[30] https://wiki.termux.com/wiki/Termux-wake-lock

- Then, download the Codiad installation script from the GitHub repository with the following command.

```
wget
https://gist.githubusercontent.com/rnauber/9f579d1480db4
cc5a9a3c97c00c52fb9/raw/install_codiad.sh
```

- After the script is downloaded, execute it with the following command.

```
bash install_codiad.sh
```

- Then, Codiad can be launched from Termux by running the "codiad" command [15].

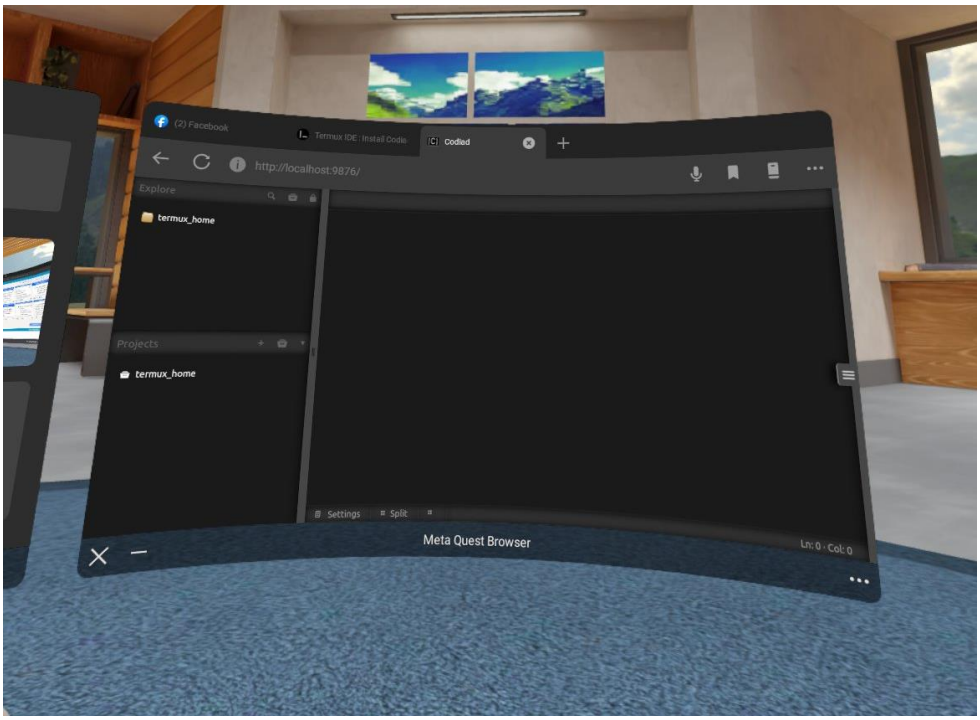

Figure 16: Codiad launched from the Termux

Codiad was launched from Termux and tried to edit elixir code with it. It seemed the code editor worked fine, but code completion was missing due to the non-availability of proper plugins.

In the Termux default environment, usage of asdf like runtime version manager is complicated, as Termux differs from Debian or other Linux distros. So, compiling a package for a Linux Distro will not work out of the box on the Termux environment[31].

Moreover, when writing this thesis, the available version of Erlang and Elixir in Termux's main repository is 25.0 and 1.13.4, compiled with Erlang/OTP 22.0. As Elixir is built on top of Erlang, the system needs to have Erlang installed, and the elixir version should be compiled with the installed version of Erlang in the system. So, currently available Erlang

[31] https://wiki.termux.com/wiki/Differences_from_Linux

and Elixir versions on the Termux package manager cannot work together unless manually compiled the elixir source with Erlang 25.0. Manually compiling the package needs to be cross compiled with Android NDK. This issue is later solved, as mentioned in section 3.2.6. Due to these issues and lack of package availability in Termux, I have tried to use proot Linux distros installed with Termux so that it is possible to get an almost full-featured Linux experience. Furthermore, the Visual Studio Code Server found in a later experiment is a much better choice to fulfill RQ4, with the possibility of using a terminal inside the code editor and many popular programming language plugins.

3.2.4 Use of Termux to install Linux on Meta Quest 2

One of our ideas was to install Linux on Meta Quest 2 and deploy the required applications and browsers on that. To install Linux on an Android device like Meta Quest 2, we have two main options.

1. Jailbreak or root the device and install Linux with chroot.
2. Install Linux on top of the android system using tools like Termux and packaged distros like proot and a viewer like VNC viewer to view it.

Jailbreaking Meta Quest is a complicated and illegal process. Furthermore, we can achieve most of the Linux features required to fulfill the requirements of this thesis without rooting Meta Quest 2. So, it is not worth the effort. However, it would be nice if Meta used Linux as their primary OS like SimulaVR[32].

On the other hand, running Linux on top of Android is easy, and lots of existing content can ease the effort. It is a legal and safe process; we will not break anything from the Meta OS.

PRoot is a program that facilitates GNu/Linux chroot-like functionality where root privilege is not available. I have used Termux to install some available proot distros with PRoot Distro's utility. *PRoot Distro* is a Bash Script wrapper utility to manage chroot-based Linux distribution installations easily. It supports almost all the latest popular Linux distros[33].

I have installed Debian and Ubuntu from PRoot Distros. Installing the PRoot distro and the distribution is straightforward, and managing distros was quite convenient. Commands to install PRoot distros and a Linux distribution with Termux are as follows:

```
pkg install proot-distro
proot-distro install debian
```

After installing the Linux distribution, we have to install a window manager. In this case, I used XFCE. AnLinux-Resources[34] on Github contains scripts to make the XFCE installation easier. So, I had to download and run the script, and XFCE is installed on our proot distro, in this case, Debian. This script also installs Xtightvnc. So, when installation is completed, the VNC server needs to stop and start with the 'vncserver-stop' and vncserver-start' commands. It will create the VNC server on localhost:1 by default.

To view and use the Debian XFCE graphical interface, a VNCViewer needs to be installed on Meta Quest 2. I had sideloaded VNC Viewer and connected to the VNCServer with it.

[32] https://simulavr.com/
[33] https://github.com/termux/proot-distro
[34] https://raw.githubusercontent.com/EXALAB/AnLinux-Resources/master/Scripts/DesktopEnvironment/Apt/Xfce4/de-apt-xfce4.sh

And XFCE desktop was displayed on the VNC Viewer at Quest 2. Installation and initial experience can be watched on YouTube [16].

Initial experience with the XFCE desktop with VNC viewer was not favorable; mouse click was not working well, the display seemed burry, and sometimes the oculus system shakes and realigns all windows. Therefore, as mentioned in the next paragraph, I have continued trying other solutions.

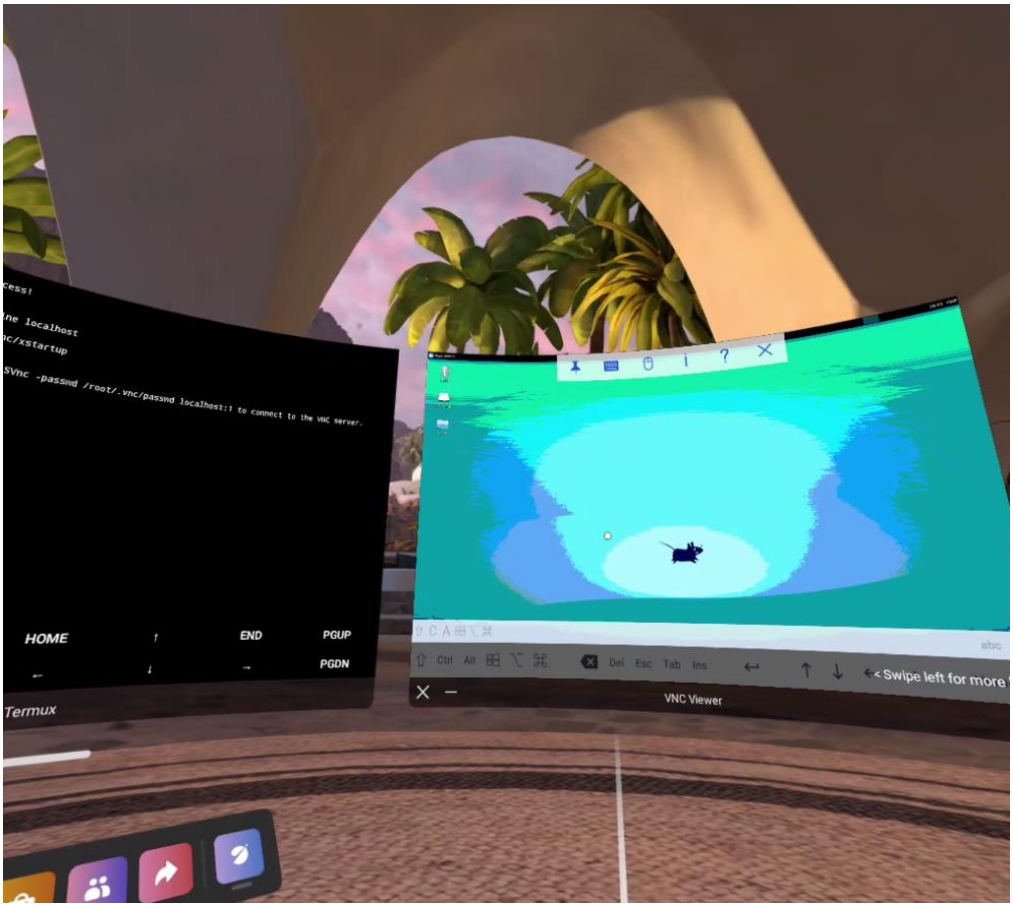

Figure 17: Debian with XFCE desktop running on Meta Quest 2

I have tried UserLand[35] and bVncViewer as UserLand provides preinstalled browsers, and new applications can be installed from the UserLand app. Experience with UserLand provided Ubuntu desktop was relatively better.

With UserLand, it was possible to browse with preinstalled Firefox graphically, and YouTube videos were played with slight lags and without audio. However, the followings are key issues while experiencing VNC Viewer-based Linux desktop.

- Keyboard and mouse not working as it supposed to on Linux environment, for copy-pasting not worked with Ctrl + Shift + C and V.
- YouTube video resolution was not good
- I had faced some audio playback issues, which were solved later with the pulse audio server. However, audio was still not a good experience due to lagging.

[35] https://userland.tech/

3.2.5 Running local Visual Studio Code server on Meta Quest 2 using Termux

Any code editor that worked on android should work on Meta Quest 2. However, while experiencing some of them, including Acode[36] and Decoder[37], it was felt that this code editor was built with a focus on small mobile screen size, whereas with Quest 2, we can experience desktop-like big screen.

Therefore, I tried to run a local Visual Studio Code server with Termux and used that as a code editor, as Visual Studio Code is not yet available for Android. Installation was straightforward with the following steps.

- Start a Linux distro instance: `proot-distro login ubuntu`
- Downloading code-server and extract:

```
wget https://github.com/cdr/code-server/releases/download/v3.10.2/code-server-3.10.2-linux-arm64.tar.gz
tar -xvf ./code-server-3.10.2-linux-arm64.tar.gz
```

- Setup a password for code-server:

```
cd code-server-3.10.2-linux-arm64
cd bin
export PASSWORD="password"
```

- Running code-server: ./`code-server`

Experiencing the Visual Studio Code server version was excellent and looked like a full-featured IDE like the desktop version.

[36] https://acode.foxdebug.com/

[37] https://dcoder.tech/

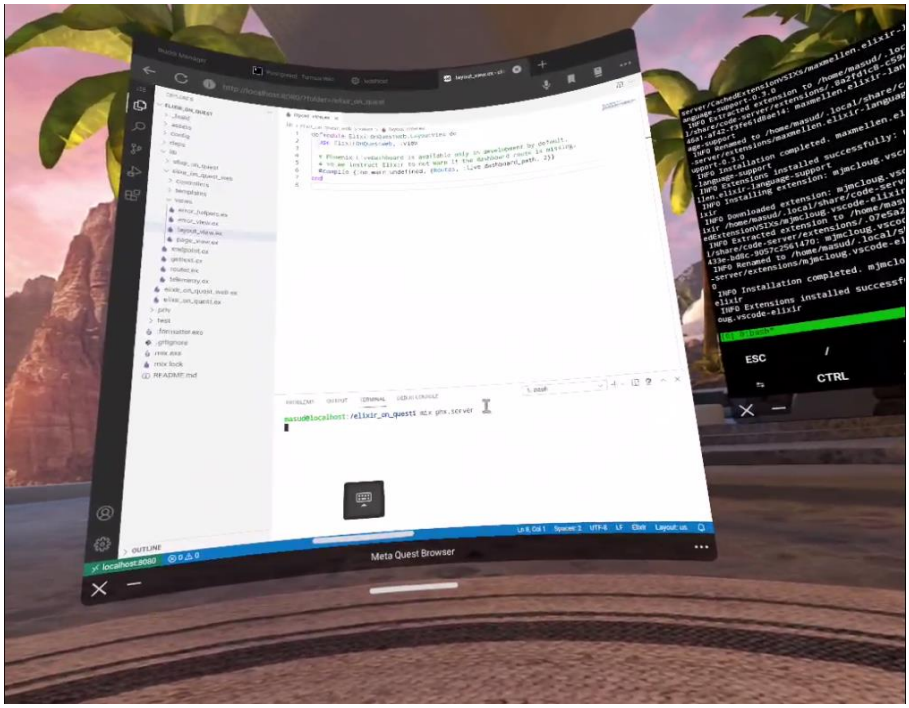

Figure 18: Visual Studio Code server on Meta Quest 2

VSCode server support plugin installation and I installed ElixirLS and VSCodeElixir. Those plugins worked as they were supposed. After the plugin installation, the VSCode page needs reloaded to get the effect. It also has terminal support. VS code server seemed the best option for code editing on Quest until now within the testing range.

3.2.6 Development experience of an elixir phoenix-based application on Termux

With the abovementioned experiments, Linux terminal emulator Termux, proot-distro Debian, and Visual Studio Code server were available to be tested with some actual application development tasks, as mentioned in RQ4. I decided to try by creating a Phoenix-based elixir application.

Phoenix[38] is a web framework written in Elixir, and it is mainly based on the model view controller pattern. This framework is built with the consideration of high performance and scalability.

Installation of Erlang, a compatible elixir version (compiled with the installed version of Erlang), and database applications like PostgreSQL are required to install Phoenix Framework and develop with it. The initial idea was to use the default Termux shell to run the Phoenix server and all the installations needed as the Termux package manager has Erlang and Elixir inside it. However, after the installation of Elixir and Erlang, it was found that they are not compatible. The installed version of Elixir is compiled with Erlang 22.0, whereas the installed version of Erlang is 25.0. This compatibility issue can be solved using a version manager like asdf to compile the required version from the source. However, with Termux, adding a version manager like asdf was not straightforward as it is not available in their package manager. It was needed to cross-compile against Android NDK, and the third-party package manager will not work on Termux as available packages

[38] https://www.phoenixframework.org/

are not compiled against Android NDK. The alternative was to use proot distro and install the required application.

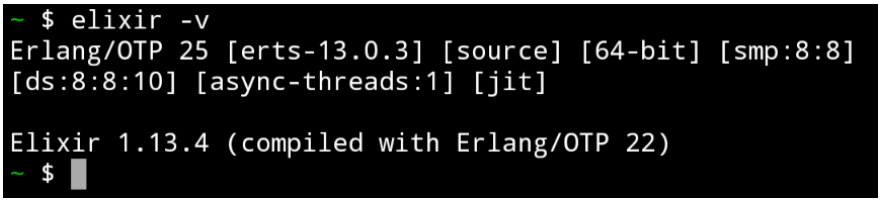

Figure 19: Elixir version conflict on Termux

Later, I logged on to the proot distro and followed the process to install erlang and elixir.

- Installed asdf as per instruction[39]
- Installed erlang 23.2.7 and elixir 1.13.3-otp-23 with asdf
- Set installed version of erlang and elixir as global

Then PostgreSQL needed to install on the system[40], and after installation of the PostgreSQL, it could not be started for the following error.

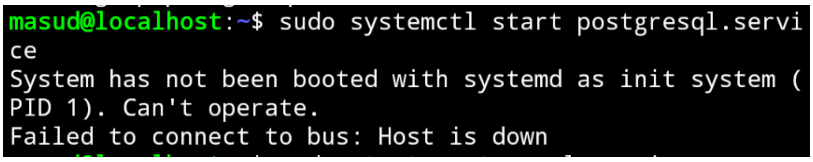

Figure 20: Error while trying to start PostgeSQL server

This error was because the system did not initialize with the init script. So, I needed to start the PostgreSQL server manually by running: `"pg_createcluster 14 main --start".` However, the server script got an ownership right error while trying to start.

After some more research, it was found that the proot system does not keep owners uid and only reports real uids. So, the workaround is to run the PostgreSQL server from the Termux default environment and connect it from the proot environment[41].

After installation of PostgreSQL logged into proot-distro again and installed Phoenix Framework. A good IDE is also needed for development, so VSCode Server is installed with the procedure mentioned in 3.2.4.

After the environment preparation was ready, I created a new phoenix project and ran the server, which worked as expected. VSCode server was also tested to install elixir-based plugins, syntax highlighting, and code changes; everything worked with a little bit of lag from the system. The complete environment preparation and instruction can be found on GitHub Project [17]. And the environment preparation experience can be found on YouTube [18]. This experiment proved that with the help of Termux, it is possible to do

[39] https://asdf-vm.com/guide/getting-started.html
[40] https://www.digitalocean.com/community/tutorials/how-to-install-postgresql-on-ubuntu-20-04-quickstart
[41] https://github.com/termux/proot/issues/116

experiments with various technologies inside the Meta Quest 2. Furthermore, it should be helpful to fulfill the goals of Pradip.

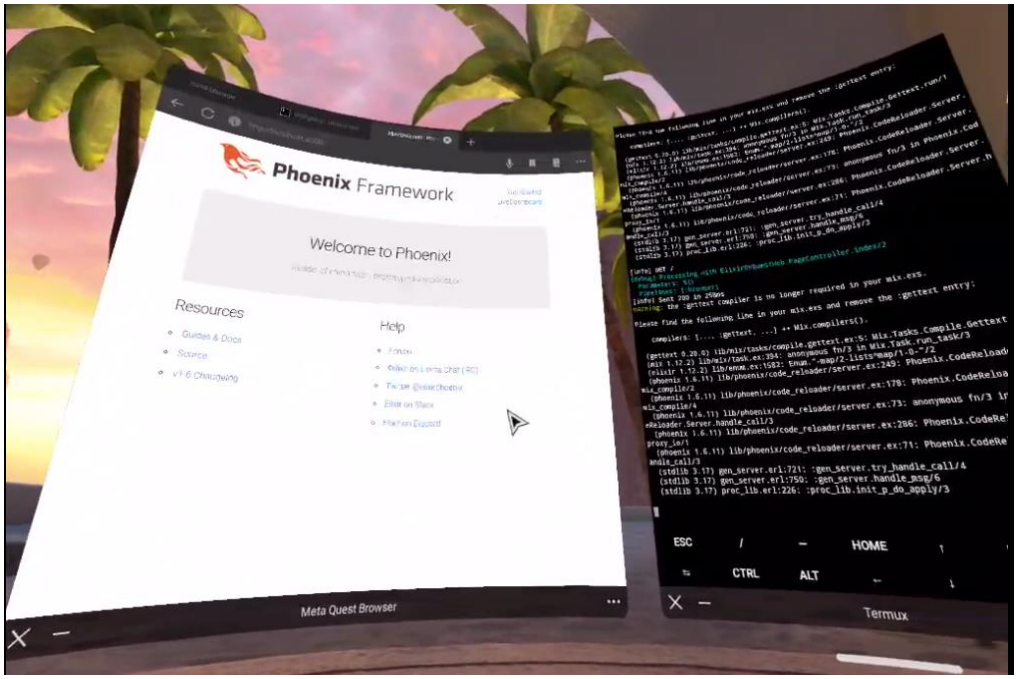

Figure 21: Phoenix server running on Quest 2 environment

Some facts were observed while experiencing Quest 2 as a Phoenix development environment, as below:

- Browser dev tools are not available for android. So, the browser console, or networking, is not available now.
- Sometimes, 2D windows are shaky, and it is recommended not to use hand tracking as it makes the experience worse.
- Copy pasting from different windows was tough. To do copy-pasting from other windows, I had to do the following steps:
    1. Initial selection with a mouse or controller by pressing the pointer on the required text
    2. Keyboard Shift + Arrow keys to precisely select the required text
    3. Keyboard Ctrl + C for copy into clipboard
    4. Hold the controller trigger button on the Termux shell or any other window to Paste.
- Virtual keyboards appeared, although a physical keyboard is attached, and it happens mainly for Termux, most probably as Termux is not from the official Meta Quest store. Moreover, Termux has some settings that allow disabling the virtual keyboard completely, and I have not tested that yet. That might be a workaround for this problem.
- Controller or mouse pointers sometimes seemed not accurate enough and made disturbance during work

The experience of developing a Phoenix-based application on a standalone Meta Quest 2 was good. The Phoenix server, VS-Code server, and PostgreSQL server ran concurrently; they consumed high system memory and processing power. So, it can be proved that Quest 2 system can handle enough load to learn basic programming, and Termux, proot-distro, and visual studio code server can be the right tool.

3.2.7 Results

Termux is an excellent tool for using most Linux packages and programming inside the VR environment. As Termux is not an approved application of Meta Store and might not be optimized for the Quest environment, it has some glitches, like a virtual keyboard appearance, although a physical keyboard is present. Except for those minor glitches, it would be a highly recommend application while education is the concern.

3.3 Development of a web-based utility tool (Buddi with phoenix elixir framework) - Work in Progress

From the requirements mentioned in section 1.2, it has been seen that RQ1, RQ2, RQ7, RQ9, and RQ10 can be achieved from our targeted Meta Quest 2 VR environment and Quest browser. The remaining requirements can be completed with different tools not solely built for that purpose, so they are not so straightforward and can break concentration while consuming learning materials. Therefore, I decided to make a tool named Buddi that will specifically solve RQ3, RQ4, RQ5, RQ7, and RQ6. RQ8 is beyond the scope of Buddi as collaborating student and teacher in the lab works virtually needed a separate application specifically built to solve only that problem.

3.3.1 Requirements selection and user stories for Buddi

As mentioned above, the purpose of Buddi is to fulfill the requirements RQ3, RQ4, RQ5, RQ7, and RQ6. Below user stories of Buddi are extracted from the targeted requirements.

1. **RQ3 - Possibility to take digital notes**: One of the critical challenges working in a VR headset is the difficulty of taking physical notes. So, digital notes are the option to take notes inside a VR headset. Digital notes can be in a different forms. The most common forms are text, voice, and images. So, Buddi should support these three notetaking forms to fulfill this requirement. Below, user stories are generated from this requirement.
    a. As a student, I want to add text notes so that I can remember the lesson I learned from the online tutorial inside the virtual reality headset.
    b. As a student, I want to update previously taken text notes so that I can add more things I have learned to them.
    c. As a student, I want to take voice notes so that I can use my voice to take notes instead of using a keyboard.
    d. As a student, I want to add an image as an image note so that I can keep the screenshot or downloaded image as a note for later use.
    e. As a student, I want to see all of the notes so that I can get an overview of the tasks I have done.
2. **RQ4 - Possibility to execute and edit basic programs:** Some sites online provide support for writing and editing simple programs online. Learning programming, writing, and executing programs can be an excellent way to start with. As one of the purposes of Buddi is to unify the student's basic needs in one place so that they will feel less disturbance while using a VR headset, this requirement can be added to the feature list of Buddi. Below, user stories are generated from this requirement.

a. As a student, I want to write and execute programs so that I can learn basic programming.
b. As a student, I want to edit previously written programs so that I can improve the program.
c. As a student, I want to see all previously written programs so that I can get an overview of my completed programming tasks.

3. **RQ5 - Writing assignment reports with basic text formatting:** Writing school assignments is one of the critical activities in the school. Currently, Google Docs[42] or Microsoft office 365[43] provides services with full-featured online document creation and manipulation tools online. Although one of the goals of Buddi is to unify the requirements, Buddi will not implement full-featured text editing and formatting tool as it's easy to use google docs instead for this specific purpose. Writing assignments is a rigorous task, and using different services is reasonable.
4. **RQ6 - Sharing files:** Sharing files is one of the everyday tasks concerning digital documents in a student's life. While inside the VR headset, it's one of the disturbing tasks as the traditional way is to connect it with another device like laptops and get files from storage or may be added USB drive with a headset to send the file into that. A digital way of sharing files is to use some file upload service like Google Drive and share the uploaded file link with the person with whom the files are intended to be shared. Although currently available digital file sharing solution is usable but needs to use different service other than Buddi, and as this is a typical task for students, this requirement is decided to be added to Buddi's feature list. Below, user stories are generated from this requirement.
    a. As a student, I want to share files with other group mates so that I can collaborate with them.
    b. As a student, I want to see my uploaded files so that I can clean and update them.
5. **RQ7 - Collaborating in group assignment:** Collaborating in a group assignment is also a typical student task, and it's closely related to RQ5. Buddi will not implement this feature now; Google Docs or Microsoft Office 365 is the most viable solution.

    So overall, the following user stories are selected for implementation:

    a. As a student, I want to add text notes so that I can remember the lesson I learned from the online tutorial inside the virtual reality headset.
    b. As a student, I want to update previously taken text notes so that I can add more things I have learned to them.
    c. As a student, I want to take voice notes so that I can use my voice to take notes instead of using a keyboard.
    d. As a student, I want to add an image as an image note so that I can keep the screenshot or downloaded image as a note for later use.
    e. As a student, I want to see all of the notes so that I can get an overview of the tasks I have done.
    f. As a student, I want to write and execute programs so that I can learn basic programming.
    g. As a student, I want to edit previously written programs so that I can improve the program.

[42] https://www.docs.google.com/
[43] https://www.office.com/

h. As a student, I want to see all previously written programs so that I can get an overview of my completed programming tasks.
i. As a student, I want to share files with other group mates so that I can collaborate with them.
j. As a student, I want to see my uploaded files so that I can clean and update them.

3.3.2 Design

To start the development of the proposed web tool first needs to decide on technology selection, basic application workflow, and user interface.

1. **Technology:** Following technologies were selected for the development of the tool Buddi.
    a. Phoenix framework: Phoenix framework[44] is an elixir-based MVC framework, and it has some excellent support with WebSocket-based live rendering with Phoenix LiveView. Requirements like taking text notes with markup support or note sharing need live UI rendering, and phoenix live view has a dedicated solution for that kind of task. Furthermore, phoenix provides a strong built-in ORM named Ecto. With Ecto, it is possible to build composable queries, write database migration, loading association, and many more excellent features without writing raw SQL queries. So, it will be used as the primary technology for development.
    b. PostgreSQL: PostgreSQL is the go-to database solution with Ecto; although we can configure it for other databases, I am sticking to it. The relational database will be used for user management and security-prone associations. However, some other storage solutions will be used for big files and media.

[44] https://www.phoenixframework.org/

2. **Application Architecture:** Application architecture follows mainly the default Model View Controller pattern provided by Phoenix Framework with slight modifications to fit Phoenix LiveView and SQL and NoSQL data storage solutions.

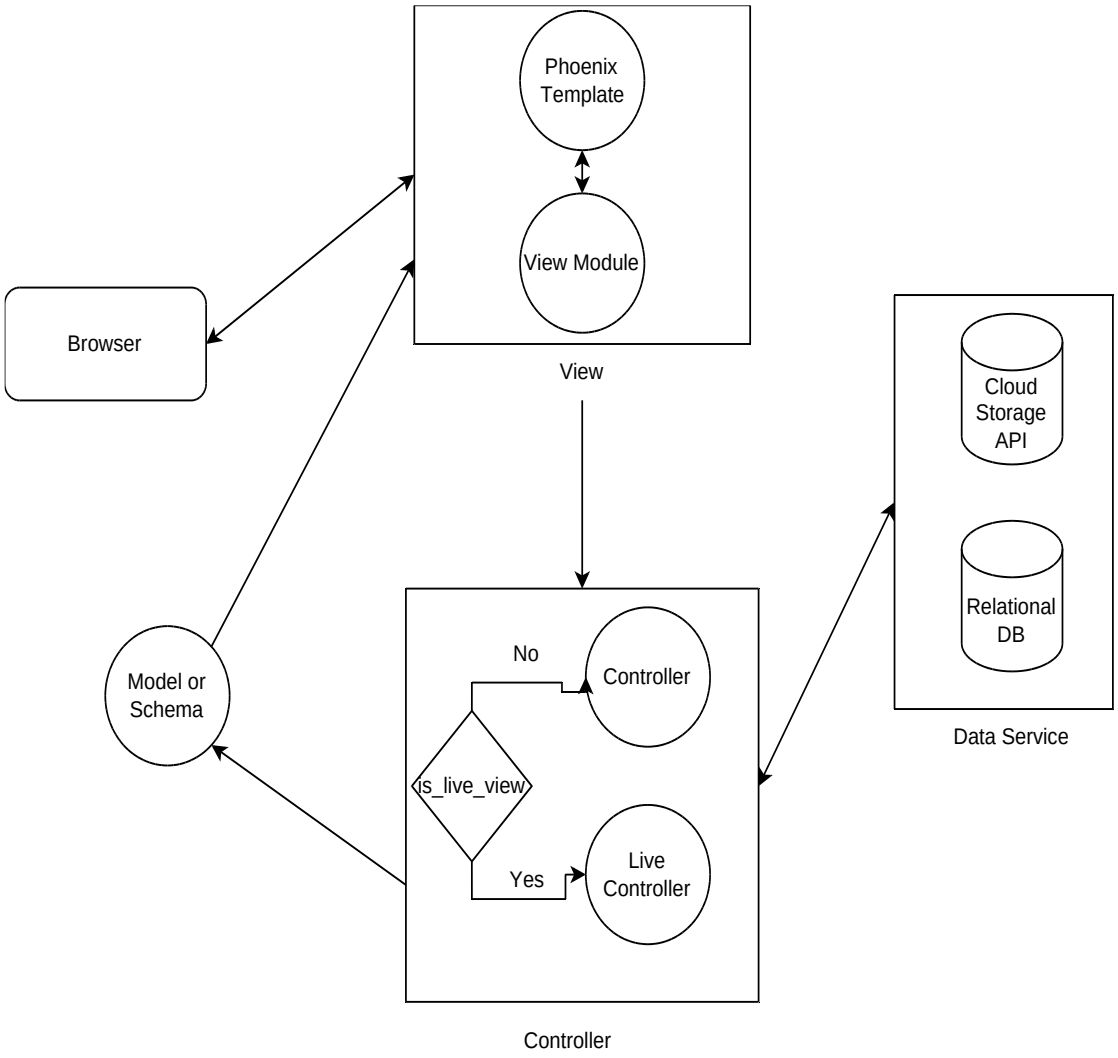

Figure 22: Application Architecture of Buddi

3. **User Interface:** Before developing some mockups, UI is made for easy visualization of the final product.

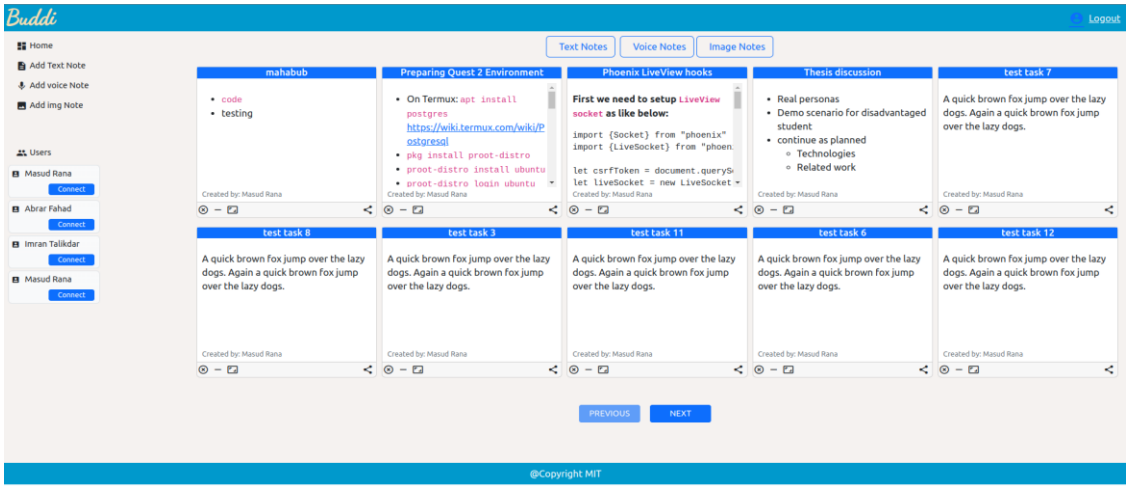

Figure 23: Application Mockup

3.3.3 Implementation

The user stories mentioned in section 3.3.1, user stories a and b, to take text notes and edit, updating them is implemented as MVP features. Implementation of other features is a work in progress. The project repository can be accessed from the reference section [19].

3.4 Summary of Experiments

Name of the Experiment	Targeted RQ	Result	Adoption to Final Guide
Godot to play videos in VR	RQ1	Success	Playing 2D videos was successful from a local source. Streaming videos can also be possible with the help of a third-party library. However, skipping this idea as fulfilling targeted RQ is possible by other means is much easier.
Enable passthrough with Godot	RQ12	Success	I skipped it as the overall idea with Godot was to unify requirements seemed complicated, and it is easy to achieve enough to show POC by other means.
Installation of Codiad with Termux	RQ4	Success	I skipped it because a better solution is installing Visual Studio Code Server with the support of a terminal and popular language plugins.
Install Linux using Proot Distros and VNCViewer	RQ1, RQ2, RQ3, RQ4, RQ5, RQ6, RQ7, RQ13	Success	I skipped it due to the issues with video sounds; the mouse and keyboard were not working as native Linux. Significantly, targeted requirements can be solved conveniently by other means.

Running Local Visual Studio Code	RQ4	Success	Adopted the guide as the solution to RQ4.
Development of Phoenix Framework-based application inside elixir	RQ4	Success	It was a test to check Meta Quest 2 systems' capability to handle serious software development. Moreover, it further proves that the system is capable of learning to program and develop software (RQ4).
Development of web-based utility tool Buddi	RQ3, RQ4, RQ5, RQ7, RQ6	-	This tool is still a work in progress. However, the text notetaking part of this tool was tested with real users and seemed successful in fulfilling RQ3. And adopted that part with the guide.

Table 1: Summary of Experiments

4 VR Learning Environment Guide

With the experiments and requirements, some guidelines and experience architecture are proposed that should fulfill most of the requirements of the targeted persona. The guidelines include applications that should be installed on the Quest 2 systems and which application needs to be used for a particular educational problem.

4.1 Hardware Setup

Using Quest 2 for educational purposes needs some basic hardware like Bluetooth/wireless keyboard, wireless mouse, and Bluetooth headphone is required. Wired versions of this hardware can be used but will not be pleasant while working in a VR environment, as per my experience while experimenting with it. I am providing a picture of my used setup as a reference below.

Figure 21: Hardware Setup for Educational Application

4.2 Preparing Meta Quest 2 Environment

To prepare Meta Quest 2 for day-to-day educational application, I would suggest installing some software we have got from our experiments. We can install software or applications from the official Meta Store, SideQuest, or Sideloading android package with the side loader application. In all these three sources, we can get android applications, but we cannot get most of the applications built targeting desktop environments like Linux, Windows, or Mac. However, as the desktop environment is mature for educational applications and already has a good ecosystem, it would be nice if we could get access to them from our Meta Quest 2. Termux and Proot distros can be handy for installing the Linux packages as we have done our experiments in section 3.2.

Below I am providing steps to prepare the Meta Quest 2 Environment:

1. Install SideQuest and Enable Developer Mode in your Quest 2 following the instruction provided on SideQuest Official Site[45] and Oculus Site[46].
2. Install Termux to the Meta Quest 2 by downloading it from F-Droid and sideloading it with SideQuest
3. Open up Termux and install a Proot distro of your choice following the instruction in section 3.2.
4. Install Visual Studio Code with the instruction provided in section 3.2.5.

With the upper-mentioned steps, it is now possible to use 2D android applications, Linux applications, official applications from the Oculus store, and unofficial VR applications from the SideQuest store. Therefore, it would be much easier for the student to fulfill specific educational needs inside the Meta Quest 2.

4.3 Use-Case Guidelines

As we have seen during the requirements extraction in section 1.2, the most common tasks for student persona are to do:

- online research to learn additional materials,
- attend online classes,
- and work to prepare assignments.

Quest 2 environment can be prepared for all the tasks mentioned above with a 2D browser window for surfing websites and doing the tasks and another browser window for taking notes with applications like Buddi. For example, while performing online research or online classes taking notes could be handy, and web applications like Buddi can help with that. Preparing assignments can be done on the browser with the help of services like Google Docs[47].

[45] https://sidequestvr.com/setup-howto
[46] https://developer.oculus.com/documentation/native/android/mobile-device-setup/
[47] https://docs.google.com/

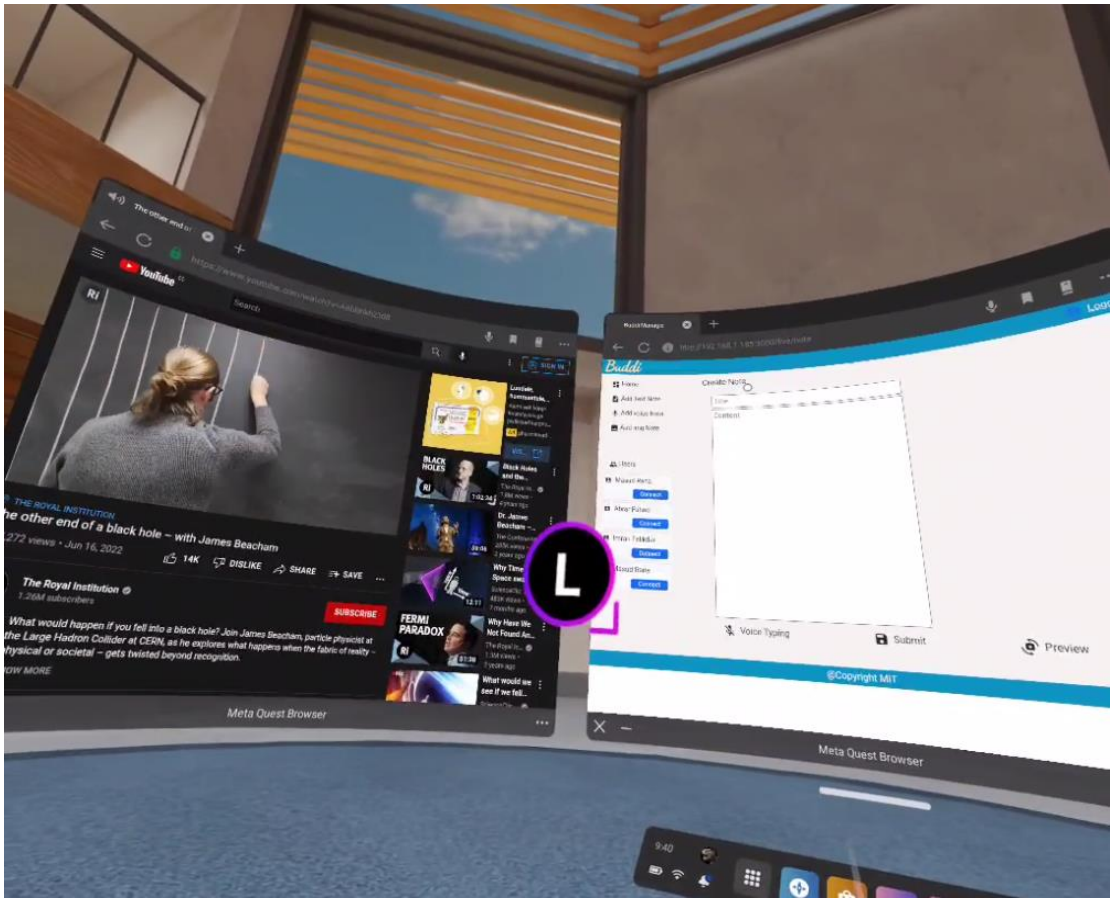

Figure 22: Watching learning videos and taking notes

Sometimes, it is needed to install some specific application to experiment with; in that case, if that particular application is unavailable in Meta Store, it can be sideloaded by downloading as apk from some sources, or the Linux version can be installed using Termux as mentioned in section 3.2.

5 Testing Experience on Quest 2

The guide provided in section 4 was tested with three people. They had been provided the following two tasks:

1. Use the provided guide to learn about the black hole inside the Meta Quest 2 environment and take text notes using the utility tool Buddi.
2. Use the provided guide and write Python code that will take one integer input(n) and print n number of elements, where each element is the incremental number of 'a'. For example:

   If the input is 3, the output should be: 'a aa aaa'

Two people took the first task, and one person took 2nd task. After they had completed their testing, a survey was conducted to understand their experiences.

All testers think solving the provided tasks with the guide is possible. Testers with task1 believe that multiple large windows helped them to learn about black holes. However, one of them faced difficulties while typing.

Tester with task 2 faced some issues setting up the editor and believed that the out-of-the-box code editor might make it much more convenient.

All three testers feel that it is possible to accomplish their tasks with the provided guide, and they made encouraging statements about it. I am quoting one of the statements as below:

"The environment provided by VR and the large screens made it easier to understand about Black holes. It was fine and made me feel as if I was really present in the lectures, videos that were presented. Even if one does not have any knowledge, I believe the VR environment can help cut off distraction and enable learning."

In future work, more users can test this guide to gather more feedback and experience, as stated in section 6.3. The reader of this thesis can access the post-survey data of the test from the URL provided in the reference [20].

So, in general, the testing experience of the provided guide was successful. Testers were excited and enjoyed testing VR environments for learning purposes and shared their discomfort while typing with the physical keyboard.

6 Conclusion

This thesis aimed to provide a practical guide to making the targeted Meta Quest 2 virtual reality environment more accessible for education. I have used user personas and respective user stories; I started extracting the requirements but also started concluding various experiments to find right technologies to fulfill the requirements and finally prepared the target guide. It also illustrates the required experiments, development, preparation, and user testing steps to provide such a guide. In this section, I will go through the requirements again to illustrate the proposed solution, share my reflection while working on this thesis, and finally, present some future work to improve the outcome of this thesis.

RQ1 - Video streaming from websites to watch tutorials or lectures: This requirement is fulfilled with the Oculus Browser. The Oculus browser supports multiple browser tabs and can be opened in different windows, making it easy to perform parallel tasks.

RQ2 - Reading digital materials like pdf or doc: Reading digital materials like pdf or doc offline needs a corresponding reader application. The guide in section 4.3 mentioned that specific applications like the reader, in this case, can be sideloaded as Apk, or if the android version of that application is unavailable, then Termux can be used to install the Linux version of that application. If an online solution is needed, then Oculus Browser can be used to consume services like Google Docs to read and edit documents.

RQ3 - Taking digital notes while consuming learning materials: The online notetaking application named Buddi is under development to solve this requirement efficiently. Furthermore, as VS code server application is proposed to install in section 4.2, it can be an offline notetaking solution. This requirement can also be solved by installing a 2D note-taker android application by sideloading it.

RQ4 - Writing and editing computer programs: VS Code server running on Termux is the solution to this requirement. Furthermore, if specific programming packages need to be installed, then Termux proot distros can be used as described in section 3.2.6.

RQ5 - Writing digital assignment reports: It can be done online with Oculus Browser and a consuming service like Google Docs or Microsoft Office 365. The offline solution to this requirement can be sideloading a document processor applications Apk.

RQ6 - Sharing files: Sharing files could be solved with the work-in-progress application Buddi. For now, it is also possible to share files using services like Google Drive with the help of Oculus Browser.

RQ7 - Collaborating in a group assignment: Collaborating in a group assignment is possible online with Oculus Browser using Google Docs services.

RQ8 - Lively online classes: Applications like Immersed or Meta Horizon Workrooms can be the solution to this requirement. Furthermore, Oculus Quest Store has these applications.

RQ9 - Collaboration with students and teachers in the online labs: Lab works differ based on topics and need a specific solution. So, learners can search for the required application to collaborate and perform the particular lab application. It is possible to search it in the Oculus Quest Store, SideQuest, or any other VR application store that can be sideloaded into Meta Quest 2. This requirement is still open. Maybe the new account system introduced by Meta could simplify this. However, this point should not be ignored in follow-up projects.

RQ10 - Visiting different places to learn Geography and History online: This requirement can be solved with Meta Quest 2 default environment by installing an application like YouTube VR from Meta Store.

RQ11 - Multitasking with multiple screens: With Meta Quest 2 default system, it is possible to open multiple windows that act like multiple screens, except multiple things cannot be opened together in one window. However, it should help to solve the multitasking requirements as with 360-degree space; many windows can be fitted. Termux with tmux installed can be one option for multitasking, while the command line is the concern.

RQ12 - Render the physical keyboard as a virtual object with keys overlay and passthrough in the keyboard area: This is possible with Meta Oculus Quest 2 system for some specific keyboards. Such an input system (and making it as affordable as possible) is, of course crucial for a digital learning environment. There are lots of changes currently happening regarding that in the Meta ecosystem, so a more general solution in the future can be very likely.

RQ13 - Intuitive operating system with near zero learning curve: Our targeted Meta Quest 2 operating system is intuitive like an android device. It should fulfill this requirement.

RQ14 - 2D app overlay within the 3D application: As Meta Quest 2 system is not open-source, to get this kind of system feature, we have to rely on the development team of Meta. Such a feature is a vital requirement, and currently, it is possible to use the universal menu from other 3D apps, so it is likely that soon Meta will add this feature in Quest 2.

So, besides requirements RQ9, RQ12 & RQ14, other requirements can be fulfilled with the provided guide and work-in-progress utility tool Buddi.

The thesis work let me cover many learning-related technologies. Termux stood out a bit as it allowed implementing anything Linux-related stand-alone on the meta quest 2, unlocking many of the solutions described above. Working with VR development in Godot faced technical challenges like selecting a proper plugin and workflow to efficiently debug the developed project due to a lack of familiarity with the VR environment, game development, improper time management, and broad initial goal. I have spent around 520hrs exploring, experimenting, and developing this thesis. That can be broken down as follows, Learning Godot and Experimenting with it -120hrs, experimenting with Termux - 140hrs, exploring some other solutions – 80hrs, developing elixir-based application Buddi – 80hrs, reading and reviewing related research and application – 100hrs.

With the provided guidelines in this thesis, it is possible to accomplish all the requirements except RQ9, RQ12 & RQ13. Along with these three requirements, we have some other places to improve in the future. Firstly, the development of all the features of Buddi mentioned in section 3.3.1 needs to be completed to make it a unified utility tool for educational applications inside VR. Unifying RQ1, RQ2, RQ3, RQ5, RQ6, RQ7, RQ11, RQ12, RQ13 & RQ14 with one 3D application might improve the system's usability in the educational context. Furthermore, it would be nice if the proposed guide is tested with more students to get more opinions and improvement opportunities.

The guide allows to replicate my experiments and experiences in the broader classroom environment. The environment has been tested successfully with positive feedback from three study colleagues that gave surprisingly positive and encouraging feedback. I am confident this guide will open more opportunities to make the target Meta Quest 2 environment more education friendly.

7 References

[1] S. Khan, "What Is An Ideal Learning Environment?," 22 June 2021. [Online]. Available: https://edtechreview.in/dictionary/4776-what-is-an-ideal-learning-environment. [Accessed 3 July 2022].

[2] N. Gilbert, "FinancesOnline," 2020. [Online]. Available: https://financesonline.com/virtual-reality-statistics/. [Accessed 15 July 2022].

[3] Meta Platforms, Inc., "Horizon Workrooms for VR Remote Collaboration," Meta, 19 April 2021. [Online]. Available: https://about.fb.com/news/2021/08/introducing-horizon-workrooms-remote-collaboration-reimagined/. [Accessed 16 July 2022].

[4] W. H. Leung, B. L. Tseng, Z.-Y. Shae, F. Hendriks and T. Chen, "REALISTIC VIDEO AVATAR," in *2000 IEEE International Conference on Multimedia and Expo. ICME2000.*, New York, NY, USA, 2000.

[5] X. Jiang, C. Liu and L. Chen, "Implementation of a Project-Based 3D Virtual Learning Environment for English Language Learning," in *2010 2nd International Conference on Education Technology and Computer (ICETC)*, 2010.

[6] J. Bacon, The Art of Community: Building the New Age of Participation, O'Reilly, 2012.

[7] B. Melissa L., S. John R. , R.-L. Ana and L. Kimberly, "Teaching and learning in Second Life: “Using the Community of Inquiry (Col) model to support online instruction with graduate students in instructional technology”," *Internet and Higher Education,* pp. 84-88, 2010.

[8] A. Petrakou, "Interacting through avatars: Virtual worlds as a context for online education," *ELSEVIER,* vol. 54, no. 4, pp. 1020-1027, 2009.

[9] N. Sun, A. Feng, R. Patton and Y. Gingold, "Programmable Virtual Reality Environments," in *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, 2021.

[10] W.-K. Liou and C.-Y. Chang, "Virtual Reality Classroom Applied to Science Education," in *IEEE*, Žabljak, Montenegro, 2018.

[11] D. W. Carruth, "Virtual reality for education and workforce training," in *2017 15th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 2017.

[12] M. Rana, "Godot Oculus Plugin Test," [Online]. Available: https://github.com/masudme09/VR4EducationGodot/tree/main/Experiments/GodotOcculusPluginTest.

[13] M. Rana, "OpenVR Plugin Test," [Online]. Available: https://github.com/masudme09/VR4EducationGodot/tree/main/Experiments/OpenVRTest.

[14] M. Rana, "Godot Project," [Online]. Available: https://github.com/masudme09/VR4EducationGodot/tree/main/Project/Buddi.

[15] S. Maqsood, "Termux IDE : Install Codiad In Termux - 2020," 16 October 2020. [Online]. Available: https://www.learntermux.tech/2020/10/Codiad-Termux-IDE.html. [Accessed 18 July 2022].

[16] M. Rana, "Installation of proot based Linux on Meta Quest 2," [Online]. Available: https://youtu.be/GNtuZgvgUCU.

[17] M. Rana, "Preparing Quest Environment," [Online]. Available: https://github.com/masudme09/VR4EducationGodot/blob/main/Documentation/preparing_quest_environment.md.

[18] M. Rana, "Preparing Meta Quest 2 Environment Experinece," [Online]. Available: https://www.youtube.com/watch?v=0KMRGbRouDQ.

[19] M. Rana, "Utility tool Buddi," [Online]. Available: https://github.com/masudme09/VR4EducationGodot/tree/main/Buddi_Manger/buddi_manager.

[20] M. Rana, "Educational Application of VR Test Feedback," [Online]. Available: https://docs.google.com/spreadsheets/d/1Icgp7LZJZmNa0_UeOAPnkwy-ipK8xchbaJp8iIO-7GI/edit?usp=sharing.

[21] M. L. Burgess, J. R. Slate, A. Roja-LeBouef and K. LaPrairie, "Teaching and learning in Second Life: Using the Community of Inquiry (Col) model to support online instruction with graduate students in instructional technology, Internet and Higher Education".

License

Non-exclusive licence to reproduce thesis and make thesis public

I, **Masud Rana**,

(*author's name*)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

   1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

   1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

of my thesis

Open and Accessible Education With Virtual Reality,

(title of thesis)

supervised by Ulrich Norbisrath,

(supervisor's name)

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **05.08.2022**