Virtual Reality Multiplayer Experience with Augmented Reality Spectator

A project report submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Technology

in

Computer Science Engineering Spl. in Artificial Intelligence and Robotics

by

Sivaraghavi U.R.

21BRS1412

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING,

Vellore Institute of Technology, Chennai,

Vandalur-Kelambakkam Road,

Chennai - 600127, India.

April 2025

Declaration

I hereby declare that the report titled ***Virtual Reality Multiplayer Experience with Augmented Reality Spectator*** submitted by Sivaraghavi (21BRS1412) to the School of Computer Science and Engieering, Vellore Institute of Technology, Chennai in partial fulfillment of the requirements for the award of **Bachelor of Technology** in **Computer Science Engineering spl in Artificial Intelligence and Robotics** is a bona-fide record of the work carried out by me under the supervision of ***Dr. Graceline Jasmine S***.
I further declare that the work reported in this report, has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or University.

Sign: ______________________________

Name & Reg. No.: ______________________________

Date: ______________________________

School of Computer Science and Engineering

Certificate

This is to certify that the project report titled ***Virtual Reality Multiplayer Experience with Augmented Reality Spectator*** submitted by ***Sivaraghavi U.R.*** **(21BRS1412.)** to Vellore Institute of Technology Chennai, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering with Specialization in Artificial Intelligence Robotics** is a bona-fide work carried out under my supervision. The project report fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Supervisor

Signature: ....

Name: ....

Date:

Head of the Department

Signature: ....

Name: ....

Date:

Examiner

Signature: ....

Name: ....

Date:

(Seal of the School)

Abstract

The development of immersive and interactive virtual environments has been made possible by the quick advances in Extended Reality (XR). This study introduces a multiplayer VR/AR platform that uses shared virtual areas to promote relaxation, education, and sales. Users can join customisable virtual rooms on the platform to collaborate in real time, interact with synced items, and communicate via voice chat. The system guarantees smooth synchronization and a consistent user experience across several devices by utilizing Normcore for multiplayer networking and integrating Google ARCore for augmented reality support.

Three main use cases are supported by the project: product showrooms, meditation spaces, and classroom instruction. In the classroom mode, teachers can hold interactive training sessions virtually, allowing distant learners to interact with peers and digital objects. By offering a dynamic and engaging shopping experience, the showroom environment enables companies to display products in a virtual context, increasing customer engagement. Finally, with adjustable atmospheric features, the meditation and relaxation mode provides users with a tranquil setting intended for stress alleviation and awareness.

Before joining or establishing rooms, members can choose names and colors for their avatars, making customisation a crucial aspect of the platform. Players may see their avatars in real time thanks to mirrors in every space, which creates a feeling of presence. The environment settings for each participant can also be chosen by the room creators, guaranteeing a controlled and engaging experience. By connecting the virtual and real worlds, the Google ARCore integration allows AR users to view the VR environment from a new angle.

Conventional multiplayer solutions like Photon and Netcode for GameObjects were taken into consideration, but they presented serious difficulties because of their complicated global server setup, lack of predefined resources, and inadequate documentation. With its integrated templates, comprehensive documentation, and effective real-time networking features, Normcore turned out to be the best option for this project. The platform makes use of XR Simulation to guarantee device compatibility and ParrelSync for multiplayer testing.

Beyond just gaming and entertainment, this VR/AR platform has important applications in business, education, and mental health. The system transforms product demonstrations, improves learning experiences, and encourages new relaxation techniques by fusing immersive technology with real-time collaboration. To further increase the system's capabilities, future updates will include AI-driven interactions, improved AR compatibility, and support for more hardware devices.

This thesis offers a thorough examination of the problems encountered, the solutions put in place, and the development process. The results demonstrate how well Normcore works as a multiplayer framework for VR/AR applications and show how immersive technology can transform digital experiences in a variety of industries.

Acknowledgements

It is my pleasure to express with a deep sense of gratitude to Dr. Graceline Jasmine S, Professor, Associate Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, for her constant guidance, continual encouragement, and understanding; more than all, she taught me patience in my endeavor. My association with her is not confined to academics only, but it is a great opportunity on my part to work with an intellectual and expert in the field of Unity Development.

It is with gratitude that I would like to extend my thanks to the visionary leader Dr. G. Viswanathan, our Honorable Chancellor, Mr. Sankar Viswanathan, Dr. Sekar Viswanathan, Dr. G. V. Selvam, Vice Presidents, Dr. Sandhya Pentareddy, Executive Director, Ms. Kadhambari S. Viswanathan, Assistant Vice-President, Dr. V. S. Kanchana Bhaaskaran, Vice-Chancellor, Dr. T. Thyagarajan, Pro-Vice Chancellor, VIT Chennai, and Dr. P. K. Manoharan, Additional Registrar, for providing an exceptional working environment and inspiring all of us during the tenure of the course.

Special mention to Dr. Ganesan R, Dean, Dr. Parvathi R, Associate Dean Academics, Dr. Geetha S, Associate Dean Research, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, for spending their valuable time and efforts in sharing their knowledge and for helping us in every aspect.

In a jubilant state, I express ingeniously my whole-hearted thanks to Dr. Harini S, Head of the Department, B.Tech. CSE with Specialization in Artificial Intelligence and Robotics, and the Project Coordinators for their valuable support and encouragement to take up and complete the thesis.

My sincere thanks to all the faculties and staff at Vellore Institute of Technology, Chennai, who helped me acquire the requisite knowledge. I would like to thank my parents for their support. It is indeed a pleasure to thank my friends who encouraged me to take up and complete this task.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

Over the last ten years, the development of virtual and augmented reality (VR/AR) technologies has drastically changed interactive experiences. Immersion systems have evolved from simple 3D simulations to complex multi-user environments that enable real-time collaboration and communication. These advancements have transformed gaming and expanded into applications for sales, training, education, and mental wellness.[1]

Industries worldwide are utilising VR/AR solutions to produce captivating, immersive experiences that blur the boundaries between the virtual and physical worlds. The proliferation of consumer-grade devices and increased processing power has made these technologies more accessible, which has led to an explosion in research and development focused on improving user interaction. This shift has led to the development of platforms that facilitate the seamless integration of different types of interaction.

This project brings together Unity, Normcore, and Google ARCore to create a multiplayer system where VR and AR users can share the same space. Whether fully immersed in VR or viewing the world through AR, users can interact with each other and their surroundings in real time. This kind of integration is especially useful for things like collaborative training, product demos, or any scenario where different perspectives matter.

At the heart of it all is real-time synchronization. Users can customize their avatars, join virtual rooms, and interact with dynamic environments, with their actions instantly reflected across all devices. Features like mirrors for self-viewing, voice chat, and interactive objects help create a seamless experience, making sure that everyone—whether in VR or AR—feels fully present and engaged.

One of the biggest challenges with cross-platform systems is making everything work smoothly without a complicated setup. This project tackles that by combining strong networking with an intuitive design. Instead of requiring a bunch of manual configurations, the system is built with ready-to-use templates and clear documentation, making it easier to scale to different user groups.

Flexibility is another key focus. The platform is designed to support a wide range of applications—virtual classrooms, product showcases, even immersive meditation spaces. No matter the use case, the core features of real-time data sync and interactive environments make it a versatile tool for creating meaningful XR experiences.

Ultimately, this project is all about breaking down barriers between VR and AR, creating a shared space where people can connect, collaborate, and engage in new ways. By pushing the limits of current tech, it opens up fresh possibilities for digital interaction and communication—things that could change the way we work, learn, and experience the world around us.

1.2 Motivation and Context

Across many industries, but especially in training, sales, and education, there is a growing need for immersive and interactive digital environments. Traditional VR solutions, which are primarily intended for fully immersive experiences, frequently fall short in accommodating the external perspectives that augmented reality (AR) offers as companies and consumers look for more engaging digital interactions. This disparity emphasizes the necessity of a single system that combines VR and AR technologies in a seamless manner.

Despite the fact that VR applications have advanced considerably, little is known about the possible advantages of combining VR and AR. These days, a lot of organizations look for solutions that let participants who are located far away not only view but also engage with digital content. A system that enables real-time, cross-platform interaction is necessary, as current platforms usually fail to close the gap between VR and AR users.

There are additional difficulties with multiplayer networking because many existing frameworks, like Netcode for GameObjects and Photon, have scalability issues and inadequate documentation. Setting up global servers and guaranteeing low-latency performance across a variety of hardware present challenges for developers. By utilizing Normcore, a well-documented framework created to streamline real-time synchronization in multiplayer settings, this project tackles these problems.

Beyond the technical difficulties, this project is further justified by the growing significance of cross-platform collaboration. Both fully immersed VR users and AR-based observers must be able to interact meaningfully in applications like online education, remote training, and virtual sales presentations. Allowing AR users to watch and interact with virtual content improves the user experience in general and encourages more lively online interactions.

Additionally, a lot of the multiplayer VR platforms that are currently available have recurring problems like uneven object synchronization and voice communication lag. These issues, which have been extensively covered in the literature (Jerald, 2015), point to the need for a more reliable solution that successfully combines AR and VR into a single, integrated system.

In conclusion, a combination of technical difficulties, market demand, and the shortcomings of current solutions are what are driving this project. It seeks to close these gaps in order to develop a more scalable and inclusive immersive platform that can support a wide range of applications in different industries. The project aims to improve user satisfaction and engagement in addition to meeting technical requirements, illustrating the wider effects of combining VR and AR technologies into a single system.

1.3 Research Challenges and Gaps

Incorporating VR and AR users into a single interactive experience presents significant challenges for current multiplayer systems. Achieving real-time synchronization between devices with widely disparate input and display technologies is at the core of the problem. As a result, user experiences are frequently fragmented because actions such as object manipulation, voice communication, and environmental adjustments frequently do not align consistently across platforms.

Making sure that a VR user's actions are instantly and precisely reflected on AR devices is a major challenge. Diverse communication protocols and hardware exacerbate latency issues, resulting in delays that disrupt immersion. The lack of thorough, well-documented frameworks that can facilitate smooth cross-platform interaction—a deficiency visible in alternative approaches like Netcode and Photon—further exacerbates this difficulty.

Furthermore, environment customization and dynamic room management add even more complexity. Players must be able to enter, change, and leave rooms during a normal session without interfering with the shared state. Performance is frequently inconsistent when there is no integrated session management solution in place, particularly when there are many users or complex interactions.

Achieving low-latency voice communication is another crucial problem. Effective collaboration and immersive experiences depend on real-time audio, but existing systems usually have delays and packet losses that reduce user engagement. Networking solutions that prioritize low latency and effective data transfer are needed to address this.

The situation is made more difficult by global scalability. Many of the systems in use today have trouble handling large numbers of users or require complex configurations in order to serve a global audience. The complex process of establishing and managing global servers creates an additional barrier to the wider use of immersive technologies.

The design of the user interface is also seriously lacking. Conventional VR solutions frequently fail to accommodate AR viewers, resulting in a subpar viewing experience for those who choose not to participate. Opportunities for improved cooperation and feedback during interactive sessions are restricted by this oversight.

In conclusion, the goal of this project is to create a single multiplayer platform that successfully combines users of VR and AR in a coordinated setting. By tackling problems with latency, synchronization, session management, and scalability, the goal is to get past the drawbacks of the current systems and provide a smooth and interesting experience for all users.

1.4 Significance and Impact

Creating a single VR/AR multiplayer platform has the potential to drastically alter how we engage with virtual worlds in a variety of domains. This method rethinks digital communication and collaboration by combining fully immersive VR with contextual overlays of AR, making experiences more accessible and engaging for a variety of users.

For instance, combining AR spectatorship and VR immersion in educational settings can produce learning environments where students engage directly with virtual content and receive real-time feedback from peers and teachers. In the end, this dynamic exchange can increase comprehension and retention of difficult subjects, improving education's efficacy and accessibility.[2]

The advantages also apply to the business world. Customers could view virtual product models in contemporary product showrooms while also accessing more augmented details on their mobile devices. These dual-mode experiences could revolutionize how products are presented, increasing consumer happiness and engagement and, eventually, brand loyalty.

In a similar vein, this integrated system might provide incredibly lifelike, interactive training and simulation modules that allow students to hone their skills in safe but dynamic environments. Employees are better equipped for tasks in the real world thanks to the synchronized interactions and instant feedback, which also expedite the learning process and reduce the learning curve.

Furthermore, it is impossible to overlook how important real-time synchronization and low-latency communication are to preserving immersion. In addition to providing a strong basis for upcoming XR innovations, overcoming these technical obstacles is crucial for seamless and reliable user interactions.

This project makes a substantial contribution to the larger body of research on cross-platform integration in extended reality, in addition to its immediate practical applications. It offers insightful information and industry best practices that can stimulate fresh lines of inquiry and creative answers to the problems associated with real-time digital interaction.

In conclusion, this project has a lot of potential to improve user engagement, foster collaboration, and spur technological innovation in a variety of industries. It fills significant gaps in existing systems and opens the door for more sophisticated and inclusive immersive experiences by fusing VR and AR seamlessly.

1.5 Project Goals and Objectives

This project's main goal is to create a single multiplayer platform that unifies users of VR and AR into a common, real-time setting. The necessity to develop an immersive system that allows both kinds of users to engage with ease and improves the user experience overall is what motivates this goal. In order to accomplish this goal, a number of technological issues must be resolved while maintaining scalability and usability.

Implementing strong real-time synchronization that enables any action taken by a VR user to be immediately mirrored on AR devices is a primary goal. Utilizing Normcore's networking features will help reduce latency and provide consistent state updates across many platforms. Here, the goal is to give users a seamless, integrated experience that keeps them immersed [3].

Enabling complete room management and avatar customisation is another goal. Before entering a shared session, users should be allowed to customize their avatars by choosing names, colors, and other features. All participants should be impacted by the room builders' capacity to create and alter the virtual environment. This feature is essential for commercial and educational applications where user engagement is greatly influenced by the environment.

Another main focus is on manipulating objects in real time. VR users must be able to interact with virtual things in a way that AR viewers can see right away. Regardless of the user's platform, this goal guarantees that interactive sessions and cooperative tasks are efficient and interesting. Precise device synchronization and smooth data transfer are necessary to accomplish this.

Including voice communication with low latency is another goal. Real-time audio improves the immersive experience and is necessary for productive teamwork. In order to facilitate prompt and clear communication between all users and strengthen their sense of presence in the virtual world, the platform intends to incorporate voice chat protocols that are optimized.

Another goal is scalability, which guarantees that the platform can accommodate a sizable user base worldwide. The system is built to automatically scale and manage growing user loads without sacrificing performance by utilizing Normcore's infrastructure. This goal is essential for the platform's effective implementation in real-world situations, where participant counts can fluctuate greatly.

Finally, by utilizing programs like ParrelSync and XR Simulation, the project seeks to offer a reliable testing and simulation environment. This goal is to guarantee the platform's dependability and consistent performance across a range of devices and configurations. To put it briefly, the goals center on developing a smooth, scalable, and captivating XR experience that connects users of VR and AR.

1.6 Technical Challenges and Considerations

Achieving smooth synchronization between VR and AR users in a shared space is one of the project's biggest obstacles. It is a challenging technical issue that calls for strong networking solutions to guarantee that every movement, contact, and communication is precisely mirrored in real time across many devices. Reduced immersion and fragmented user experiences might result from inconsistent data synchronization.

Another significant problem is latency in object manipulation and speech communication. In addition to interfering with the flow of exchanges, high latency also reduces the sense of presence, which is necessary for productive teamwork. Due to scalability problems and inadequate documentation, existing technologies such as Netcode and Photon have had difficulty meeting these real-time performance criteria. The project's success depends on overcoming these obstacles.

Managing changeable room configurations and user sessions presents additional difficulties for the project. A complex backend system is required to enable users to establish or join rooms, alter the environment in real time, and modify their avatars. Real-time data changes, session persistence, and user authentication must all be handled by this system with dependability and without interruption or inconsistency.

Another level of complication arises when integrating various technologies. The platform integrates Google ARCore for augmented reality, Normcore for networking, and Unity's XR framework, each of which has specific requirements and protocols. A major technical challenge that calls for careful planning and strong integration techniques is making sure that these heterogeneous systems function together seamlessly.

Additionally, controlling scalability is a big worry. Practical applications require low latency support for a worldwide user base, however many existing systems fall short in this regard. In order to guarantee that performance stays constant as the number of users grows, the project needs to handle problems with server capacity, load balancing, and network bandwidth.

There are also several difficulties with user experience, especially when it comes to creating user-friendly interfaces that work for both VR and AR users. AR users have a less interesting experience because traditional VR environments are not designed with spectatorship in mind. Innovative UI/UX solutions that offer a smooth, consistent experience for every user, irrespective of their interaction method, are needed to solve this issue.

All things considered, the project difficulty includes a wide variety of technical and user-related difficulties. To create the intended immersive experience, every component of the system needs to be precisely calibrated, from low-latency communication and real-time synchronization to seamless integration and scalability. This complex issue emphasizes the need for a fresh, well-researched solution—one that takes advantage of Normcore's advantages while resolving the fundamental drawbacks of earlier methods.

1.7 Project Scope and Future Directions

The creation of a comprehensive, cross-platform multiplayer system that unifies VR and AR players in a communal, interactive setting is included in the project's scope. The project's technological stack consists of Google ARCore to enable augmented reality features, Normcore for real-time networking, and Unity for creating the immersive environment. Blender is also utilized to create unique environmental assets that improve the platform's aesthetic appeal.

Product showrooms for improved retail presentations, virtual classrooms for training and education, and meditation spaces for mental and physical well-being are just a few of the use cases that the project is made to accommodate. The fundamental features of the system, including dynamic room customisation, real-time synchronization, and interactive object manipulation, are carefully included into each of these scenarios. This wide range of applications highlights the platform's adaptability.

The project uses technologies like XR Simulation for laptop-based testing and ParrelSync for local multiplayer testing to guarantee reliable multiplayer functionality. These testing techniques are essential for verifying how well the system performs in diverse situations and making sure that the user experience is consistent across hardware configurations. Thorough testing procedures aid in locating and resolving such problems prior to widespread implementation.

In the project scope, global scalability is a crucial component. The infrastructure of Normcore offers the support required for low-latency communication and real-time data synchronization, and the system is designed to accommodate thousands of concurrent users. Deploying the platform in real-world settings, where user counts can vary greatly, requires this scalability.

The project also discusses how to integrate various platforms and technologies. The creation of a cohesive framework capable of efficiently managing and synchronizing diverse data streams is necessary for the integration of Unity, Normcore, and Google ARCore. To make sure that every component functions as a whole, this part of the project requires thorough study of networking protocols and complex system design.

Additionally, a thorough assessment of the user experience in various scenarios is part of the scope. The system's performance is evaluated by the measurement and analysis of metrics like latency, synchronization accuracy, and user engagement. This assessment highlights areas for future development and offers insightful information about the platform's efficacy.

Lastly, even if the system's development and implementation are the main priorities, the project also takes potential future growth into account. These include adding more XR devices, utilizing AI-powered analytics to analyze user activity, and improving the platform's scalability even more. The project's extensive scope lays the groundwork for further study and advancement in the field of immersive, cross-platform interaction.

Chapter 2

Literature Survey

2.1 Cross-Platform AR/VR Development: Strategies for Seamless User Experiences

Maharani and Das[4] provide a comprehensive exploration of cross-platform AR/VR development, highlighting significant challenges such as hardware fragmentation, inconsistent UI/UX adaptations, and performance bottlenecks due to reliance on platform-specific SDKs and limited real-time data synchronization. Their work reveals that these issues often result in degraded user experiences, particularly on lower-end devices, and hamper the scalability of immersive applications. In contrast, our project leverages Unity, XR Management, and Normcore to create a hybrid VR-AR system that specifically addresses these challenges. By integrating a real-time multiplayer VR environment with an AR spectator mode, our approach enhances synchronization and reduces latency, thereby overcoming the limitations noted by Maharani and Das[4]. This not only improves engagement and interactivity but also sets the stage for further innovations in adaptive rendering and AI-driven optimization for cross-platform AR/VR solutions.

2.2 Designing in Virtual Reality

In their exploration of virtual reality's (VR) revolutionary potential in design processes, Di Gironimo and Lanzotti [5] highlight VR's capacity to produce immersive and interactive environments for ideation and assessment of new concepts. Their work demonstrates how virtual reality (VR) may be used in industries including automobile engineering, architecture, and manufacturing, where it allows designers to create very realistic simulations of human interactions. Nonetheless, the authors point out several important drawbacks, such as the requirement for more accurate user behaviour models, replicable virtual trials, and improved VR interface with current CAD software. These gaps highlight the difficulties in producing realistic and captivating virtual reality experiences, which makes them especially pertinent to our research. Our approach addresses these problems by integrating Augmented Reality (AR) spectatorship into virtual reality (VR) multiplayer scenarios.[5] This enables AR users to watch and engage with VR participants in real-time. The authors point out that the problem of unpredictability in virtual studies can be addressed by utilizing Normcore for networking, which guarantees synchronous interactions between AR and VR users.

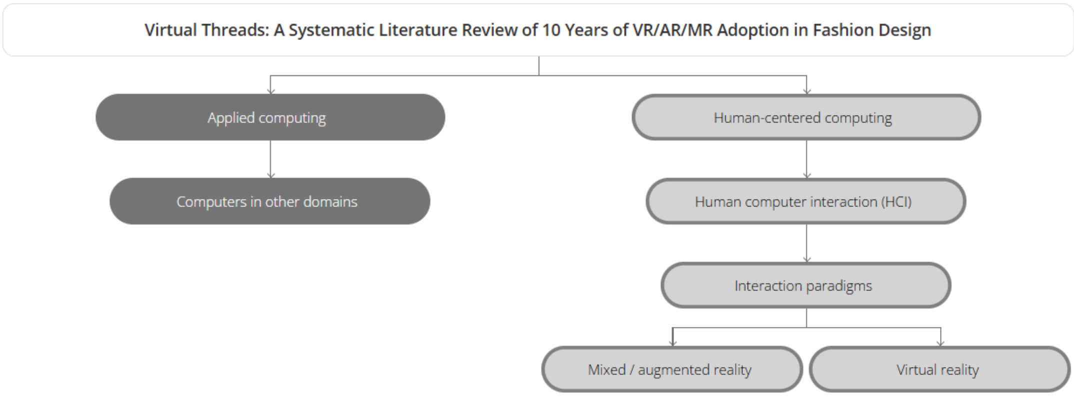

FIGURE 2.1: Review of 10 Years of VR/AR/MR Adoption in Fashion Design

Additionally, our project's objective of increasing engagement through immersive experiences is in line with the paper's emphasis on user-centered design. Compared to conventional VR setups, our solution records user feedback and behaviour in a more dynamic and participatory way by allowing AR spectators to take part in VR training or sales scenarios. As envisioned by Di Gironimo and Lanzotti,[5] this method improves realism, encourages teamwork, and advances new design paradigms that emphasize real-time interaction and user involvement. Although their study delivers insightful information about the potential of VR for design applications, its shortcomings in terms of system integration and user interface realism point to areas where our project makes a unique contribution. By bridging the gap between VR interactivity and AR spectatorship, our work extends the applicability of immersive technologies to domains such as training, teaching, and sales, directly addressing some of the key challenges identified in this study.

2.3 Comparison of Client-Server Solutions in the Development of Massively Multiplayer Online Games on Unity

In their critical analysis of typical Unity multiplayer game development, Mukhametkhanov, Khafizov, and Shubin[6] draw attention to the communication and scalability issues with client-server architectures. They suggest utilizing Golang to replace the Mirror library with a microservices-based strategy, highlighting its benefits in managing heavy user loads and accelerating message delivery. [6] Although the study offers compelling proof that integrating Unity and Golang improves system responsiveness, it does not thoroughly examine how it affects user engagement, which is an important consideration in multiplayer settings. Within this framework, the authors also go over debugging methods for multi-threaded applications.

Our concept, which combines a microservices architecture with Normcore for networking, is directly informed by this research and improves real-time interactions between VR users and AR viewers. Although they concentrate on technological performance, our study goes beyond their conclusions by giving user involvement and system integration realism top priority. We hope to solve gaps in user interaction realism that their study does not completely examine by utilizing AR spectatorship in VR training and sales environments to offer a more participatory experience.

2.4 Networking for Computer Games

In its editorial, Smed examines the critical role that networking plays in contemporary gaming, highlighting how multiplayer features in online consoles, commercial games, and mobile entertainment are all dependent on it.[7] It introduces a specific problem that looks at important networking issues, like making sure real-time interactions are responsive and consistent. A system for evaluating state fidelity in prediction techniques is presented in one study, and middleware for streamlining online game creation is covered in another. Scalability is a key concern, and potential solutions include a general framework for MMO deployment and a way to keep mobile games going in the event of network outages. The final study emphasizes the use of player behaviour analysis for cheat detection and tailored content.

Although the editorial offers insightful commentary on technological developments, it mostly ignores user experience and participation. By incorporating AR spectatorship into VR environments and promoting real-time interactions between VR users and AR observers, our initiative expands on these findings. Our goal is to close the gap between player happiness in multiplayer gaming and technology improvements by striking a balance between improved user experience and networking efficiency.

2.5 Virtual reality system for industrial training

To solve issues with industrial training, including exorbitant expenses, safety hazards, and the requirement for recurrent instruction in dangerous settings, Liu and Li developed a virtual reality training system.[8] Their solution incorporates multi-dimensional data input, simulates real-world manufacturing scenarios, illustrates a variety of processes, and provides interactive feedback to illustrate errors. The VR system reduces dangers while increasing the effectiveness of industrial training by offering a secure and affordable substitute. The study does have certain shortcomings, though, such as a lack of attention to user engagement tactics and a lack of metrics to assess the efficacy of training.

This study supports our project's objective of using AR spectatorship to improve real-time interactions in VR. We hope that by including AR, spectators will be able to communicate with VR users, encouraging cooperative learning and engagement that goes beyond conventional approaches. Our design decisions are influenced by Liu and Li's focus on high-fidelity simulations, which highlights the necessity of immersive surroundings. Furthermore, our research fills in the gaps in user experience and interaction quality by integrating cutting-edge networking solutions like Normcore, which eventually improves the efficacy of VR-based industrial training.

2.6 Strategies for Cross-Platform Web AR Development

Cross-platform Compared to native applications, web AR development has several benefits, especially in terms of affordability and accessibility. Without installing apps, developers may create browser-based augmented reality experiences by utilizing tools like Unity, ARKit, and ARCore. However, limited access to hardware resources limits the performance of these experiences. To overcome this difficulty, the project makes use of Normcore and Unity's XR ecosystem to enable real-time networking and synchronization between VR and AR devices. Furthermore, the initiative builds on current immersive interaction models by incorporating AR spectators into the experience, fostering a more inclusive and cooperative XR environment.[15]

A key component of XR development is multiplayer networking, with options such as Mirror, Photon PUN, and Telepathy providing a range of trade-offs. While Photon PUN has trouble scaling in big multiplayer contexts, Mirror is renowned for its simplicity but lacks strong real-time synchronization. The project uses Normcore, which is especially made for real-time state synchronization—a crucial prerequisite for interactive VR and AR applications—to get around these issues. By efficiently lowering latency and bandwidth consumption, its improved delta compression algorithms guarantee a smooth and scalable multiplayer experience.

Another crucial component of XR systems is latency optimization; studies have shown how to reduce inter-player latency using methods like edge computing. Although the project does not make use of edge computing, it maintains low latency and seamless interactions by applying comparable concepts through the decentralized architecture of Normcore. It also includes dynamic object ownership transfer, which enables smooth interaction between VR players and AR viewers without synchronization problems. This strategy is in line with previous studies on avoiding conflicting updates and stale ownership statuses in multi-user XR systems.[15]

The project incorporates Unity's XR Interaction Toolkit in addition to networking and synchronization to effectively manage multi-device input mapping. By supporting OpenXR standards and offering a prefab-based approach for object interaction and avatar customisation, this toolkit streamlines development. In order to improve realism, the project also uses Normcore's RealtimeAvatar API for real-time inverse kinematics (IK) synchronization, which guarantees precise hand and gesture duplication between players. Additionally, effective data transmission is made possible by bandwidth optimization techniques like delta compression, which lower network payloads while preserving high synchronization accuracy.

The initiative expands on previous studies on cross-reality collaboration, especially in domains like education and retail. The advantages of enabling AR spectators to watch and mentor VR users are illustrated by case studies on XR for medical education and sales training, which enhance learning results and engagement. By putting comparable ideas into practice, the initiative improves on existing practices and provides a dynamic and engaging platform for sales presentations and training. By improving real-time collaboration and increasing the potential for cross-platform interactions, these advances strengthen the project's contribution to XR development.

2.7 Molecular MR Multiplayer: a cross-platform collaborative interactive game for scientists

Augmented reality (AR) and virtual reality (VR) components are combined in mixed reality (MR), which enables users to interact with digital material in real-world settings. The study Molecular MR Multiplayer investigates how MR can revolutionize scientific collaboration, especially in domains where seeing intricate molecular structures is crucial,[16] such as chemistry and molecular biology. The authors' use of a game-based learning strategy makes complex scientific ideas more participatory and interesting, which aids researchers in better understanding them. One of their system's main advantages is its cross-platform capability, which helps to close the gap sometimes observed in scientific software by facilitating smooth user communication across various devices, including PCs, tablets, and VR headsets. Because of its inclusivity, a greater variety of experts can engage in cooperative research projects, making the platform more accessible.

Unity, which offers multiplayer support, real-time interactivity, and effective rendering of intricate molecular images, is used to construct the study's technological underpinning. With interactive capabilities that enable researchers to work together in real time and change molecular structures, a significant emphasis is placed on user involvement, which promotes deeper learning and teamwork. The study does, however, also recognize a number of difficulties,[16] such as hardware constraints, the learning curve for users who are not experienced with MR environments, and the significant work needed to produce instructional materials that are scientifically accurate. Notwithstanding these obstacles, the study identifies methods for enhancing accessibility, like enhancing software functionality and creating more intuitive user interfaces.

Multiplayer VR and AR spectator systems for instruction and sales are especially pertinent to this research. Your concept depends on synchronized object states, similar to molecular MR collaboration, in which AR viewers watch changes in real time while VR users manipulate items. Your AR spectator system, which brings virtual products into real-world settings for training sessions and sales demos, is a great fit with the idea of superimposing digital content onto the real environment. Furthermore, your project benefits from Normcore's real-time networking, which is already tuned for XR applications, even though the paper employs a unique client-server approach. Their strategy for resolving disputes—which guarantees harmonious interactions between several users—can also guide your implementation, where cooperation is facilitated by Normcore's integrated ownership transfer.

Your work expands on the Molecular MR Multiplayer paper's strong foundation for cross-reality collaboration by integrating these ideas into educational and commercial contexts. Your project utilizes OpenXR and AR Foundation, guaranteeing wider interoperability across VR, AR, and mobile platforms, in contrast to their approach, which is restricted to HoloLens and desktop configurations.[16] Your system can accommodate more users than their computationally demanding molecular rendering by utilizing Normcore's effective networking. Lastly, your interface prioritizes a straightforward user experience and is made for sales teams and educators, whereas theirs is made for scientists. As MR technology develops further, adding capabilities like AI-driven suggestions and hybrid rendering may improve immersive learning and teamwork even more, increasing the usefulness of these tools for training settings and enterprises.

Chapter 3

Methodology

3.1 System Architecture Overview

A thorough overview of the VR/AR multiplayer platform's overall system architecture is provided in this section. The User Interface, Application, Synchronization Communication, Management, and Data layers are all included in this modular, tiered design. To produce a seamless, immersive experience, scalability, low latency, and seamless synchronization are prioritized across all devices.

Definition

Through the use of a headset, **Virtual reality (VR)** immerses people in an entirely digital world, cutting them off from reality. **Augmented Reality (AR)** improves a user's perception of their environment by superimposing digital content onto the physical world, typically through smartphones or AR glasses. Digital items can interact in real time with real-world elements thanks to **Mixed Reality (MR)**, which more interactively mixes the virtual and physical worlds. The phrase **Extended reality (XR)** refers to all immersive technologies, including virtual reality, augmented reality, and mixed reality. The term **multiplayer** describes the ability of several people to communicate and take part in a common virtual or mixed environment, frequently at the same time, promoting cooperative or competitive activities.

3.1.1 Overall System Design

Incorporating immersive VR and AR experiences onto a single multiplayer platform is the goal of the system architecture. The overall design places a strong emphasis on scalability and modularity, making sure that various elements like environment management, real-time synchronization, and avatar customisation function as a whole. The design offers a versatile foundation for adding other XR features by utilizing Unity as the programming environment.

The system is separated into different layers by the architecture: the Application, Data, Management, Synchronization Communication, and User Interface layers. This division of responsibilities makes maintenance, potential growth, and troubleshooting simpler. By interacting over well-defined protocols, each layer makes sure that data flows efficiently and that modifications made to one module don't negatively impact others.

Furthermore, ensuring real-time updates across VR and AR devices and minimizing latency are important goals in the overall system design. To accomplish this, Normcore is used for networking, which manages state synchronization with minimal overhead. To ensure that all user actions are precisely mirrored across platforms, the design also integrates Google ARCore, which offers powerful AR streaming and overlay capabilities.

3.1.2 Detailed Component Architecture

Several essential modules are part of the comprehensive component architecture. The Pregame Environment Module, where users can create or join rooms, choose preferences, and alter their avatars, is at the forefront. By directly interacting with the Session Management component, this module makes sure that user information is synchronized and verified prior to accessing the main environment.

The main interactive elements, like the AR Visualization Module and the VR Environment Module, cooperate to give every user a seamless experience. While the AR Visualization Module processes and streams the VR environment to AR devices using ARCore, the VR Environment Module controls the immersive space and object interactions. The real-time synchronization system controls the data flows between these modules, guaranteeing uniformity for every client.

Additionally, the Management and Communication layers cooperate to support real-time updates and preserve session integrity. As networking's backbone, Normcore makes sure that every interaction—whether it be a voice command, object manipulation, or movement—is immediately transmitted to every device that is connected. The system as a whole is built on this intricate architecture, which facilitates smooth cross-platform communication.

3.2 Game Flow and Interaction Design

The user journey is described in this section, starting with the initial room setup and pregame customization and ending with real-time interactions in the virtual world. It describes how users can create or join rooms, customize their avatars, and use mirror feedback to visualize themselves. Throughout the session, synchronized and captivating interactions are guaranteed for both VR participants and AR viewers thanks to the flow design.

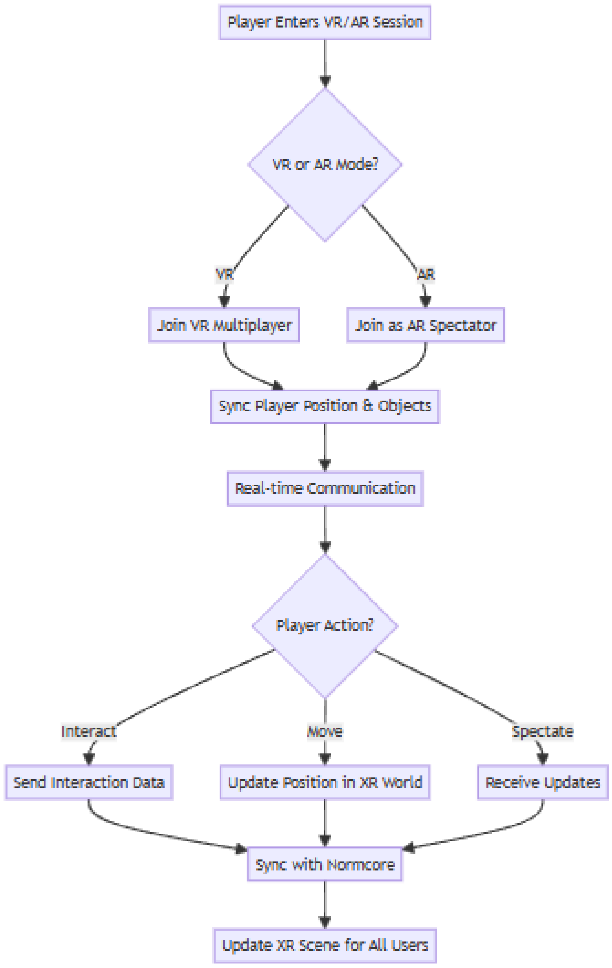

Figure 3.1: Virtual Reality and Augmented Reality Multiplayer Flowchart

3.2.1 Pre-game Environment and Customization

Before entering the main session, users are welcomed with a lobby in the pre-game environment where they can customize their experience. Users are asked to enter their names and choose an avatar color upon entry, and the system instantly records these choices. In addition to personalizing the experience, this customization process gives each network user a distinct identity.

The pre-game environment's design places a strong emphasis on ease of use and simple navigation. By entering a room name, players have the option to join an existing room or create a new one. Additional choices are offered to room creators so they can choose the theme or setting (such as a showroom, classroom, or meditation area) that will be used for all attendees. Since it creates the framework for the shared immersive experience, this step is essential.

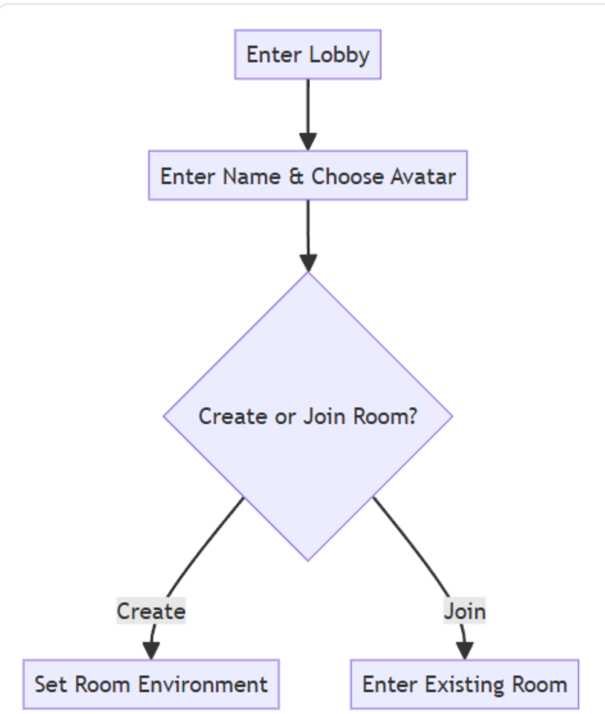

FIGURE 3.2: Pre-game Environment (Customization Room Setup)

Normcore manages network state synchronization, user session creation, and room authentication, all of which are crucial to this stage. The customization choices made by each user are consistently reflected across all devices thanks to Normcore's use of predefined templates and protocols. For the subsequent immersive experience to be built, this smooth integration in the pre-game stage is crucial.

3.2.2 Room Entry and Self-Visualization

Users enter the room where the immersive experience begins after completing the customization process. A key element of this stage is the self-visualization mechanism; mirrors are placed carefully throughout the environment to allow users to see their avatars in real time. Thanks to this immediate feedback, users can reinforce their presence in the virtual world and validate their customization choices.

In addition to self-visualization, avatar positions and appearances are updated in real-time while entering a room. Normcore synchronizes these details across all connected clients so that any change is instantly visible to both VR players and AR viewers. This synchronization is necessary to maintain immersion and consistency because discrepancies can lead to a fragmented user experience.

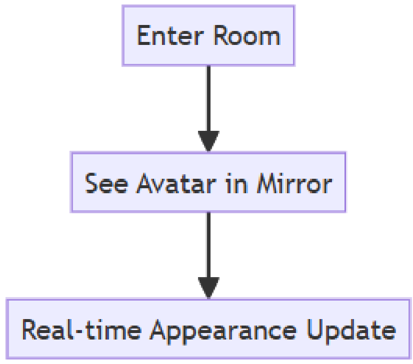

FIGURE 3.3: Proposed DI & A architecture

Strong session management, which controls user authentication and room permissions in addition to handling avatar data, supports the room entry phase. This layer ensures that only authorized users are permitted to participate and that all interactions within the room are secure. A steady, dependable entry procedure that serves as the basis for further interactive sessions is the end result.

3.2.3 Multiplayer Interaction Mechanisms

The interaction mechanisms activate once the player enters the room, enabling them to actively interact with the virtual world. In order to coordinate with others, players can move, interact with objects, and speak. Every action, like picking up or moving an object, is synchronized in real time across all clients thanks to the system's support for dynamic interactions.

The experience is customized for AR viewers to watch and, in certain situations, give feedback on the ongoing interactions. The AR module uses Google ARCore to stream the 3D environment to mobile devices so viewers can watch VR players in action. By offering several viewpoints in a single session, this dual-mode interaction greatly improves collaboration.

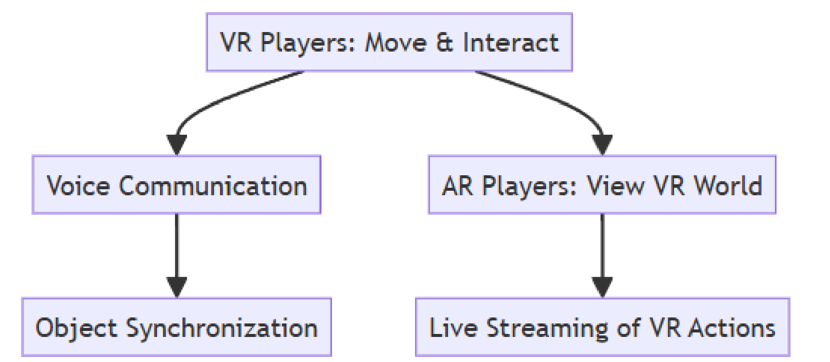

FIGURE 3.4: Multiplayer Interaction

In order to ensure that all interactions—regardless of the kind of device—are constantly synchronized, Normcore is essential. Effective data protocols are used by the system to maintain low latency, which is necessary for both successful communication and immersive engagement. Through the coordination of user actions and environmental changes, the platform creates a seamless and captivating multiplayer experience.

3.3 Networking and Real-time Synchronization

The platform's underlying real-time synchronization and networking techniques are examined in this section. Its primary goal is to integrate Normcore as the core solution for efficient state management, low-latency communication, and data replication. Cutting-edge techniques that ensure every user action—whether in AR or VR—is instantly mirrored on all devices are discussed.

3.3.1 Normcore Integration

Normcore is integrated as the main networking solution to manage real-time synchronization across the platform. Normcore was selected due to its extensive documentation, ease of use, and prebuilt templates, which significantly reduced development time when compared to other networking solutions like Netcode or Photon. Because it handles important functions like session creation, data synchronization, and latency management, Normcore is necessary to maintain an immersive experience.

To synchronize user data, object states, and environmental changes throughout the integration process, Normcore needs to be set up in the Unity environment. This configuration ensures that every user movement or interaction is instantly propagated across all connected clients. Normcore's prebuilt templates streamline this process, allowing for efficient deployment and rapid prototyping.

Furthermore, scalability is supported by the architecture of Normcore. With low latency and high reliability, it can manage several users at once. For applications where participant numbers can vary greatly, this feature is essential for maintaining platform responsiveness even under high load. One essential element supporting the system's overall performance is the integration of Normcore.

3.3.2 Real-time Data Synchronization

Whether in VR or AR mode, real-time data synchronization is a key component of the project that guarantees all users receive real-time updates on the environment's condition. The effective data replication protocols of Normcore are used to accomplish this synchronization. All user actions are recorded, processed, and sent to all other participants in real time, including moving the avatar and manipulating objects.

To ensure a consistent state across all devices, the system uses a continuous data streaming mechanism that sends updates frequently. By using this method, the chance of latency is reduced and the immersive experience is protected from interruptions. Thorough testing using ParrelSync and other tools has demonstrated that the synchronization is resilient even when there is a high network load.

The system makes use of frame similarity caching and predictive algorithms to further improve real-time performance. These methods lessen the time between data transmission and display and help predict user behavior. The outcome is a seamless, fluid experience that preserves the integrity of the shared environment by instantly mirroring one user's actions on all other devices.

3.3.3 Latency Management Strategies

For any real-time multiplayer system to be successful, latency management is essential. This project uses a number of techniques to reduce latency and guarantee seamless interactions. The low-latency networking protocols of Normcore serve as the basis, and other strategies like effective serialization and data compression help to further minimize transmission delays.

By strategically using buffering to address temporary network problems and optimizing network packet sizes, latency management is also accomplished. These techniques guarantee that the overall experience stays constant even when network conditions change.

To ensure that the system can withstand peak loads without experiencing appreciable performance degradation, extensive testing has been done to fine-tune these parameters.

Furthermore, since audio delays can ruin the immersive experience, latency must be carefully considered when integrating voice communication. To ensure timely and clear audio streams, Normcore's networking capabilities are combined with optimized voice chat protocols. Together, these techniques guarantee that audio and visual information is delivered with the least amount of latency possible, facilitating a genuinely dynamic and captivating user experience.

3.4 AR Integration with Google ARCore

This section focuses on using Google ARCore to integrate augmented reality features into the multiplayer environment. It describes how a real-time data stream and overlay mechanisms allow AR viewers to view and engage with the VR world. To provide a high-fidelity augmented reality experience, important issues like rendering quality and device adaptability are resolved.

3.4.1 AR Spectatorship Mechanism

A specialized AR module that makes use of Google ARCore is required to incorporate augmented reality into the multiplayer environment. In order for AR viewers to experience the VR session from a different angle, this module is in charge of recording and rendering the virtual environment on mobile devices. This is made possible by the ARCore framework, which offers strong motion sensing, environmental awareness, and tracking features.

By superimposing the virtual world in real time on the user's physical surroundings, the AR spectatorship mechanism is intended to enhance the VR experience. Users of AR can watch live interactions and get updates on the status of the virtual environment thanks to this integration. To maintain a consistent and engaging experience, the system makes sure that any modifications made to the VR environment are instantly reflected in the AR view.

The AR module is also performance-optimized across a variety of mobile devices. Particular care is taken to guarantee that the system's overall responsiveness is not jeopardized by the AR rendering. The platform offers a high-quality, low-latency augmented reality experience that seamlessly blends with the VR elements by utilizing ARCore's sophisticated features.

3.4.2 AR Streaming and Overlay Technologies

A complex overlay mechanism that integrates real-time data from Normcore with ARCore's rendering engine is used to stream the VR environment to AR devices. Thanks to this technology, viewers of AR can see a real-time representation of the VR environment, complete with moving avatars, dynamic objects, and shifting surroundings. By correctly aligning virtual elements with the real world, the overlay technology creates a seamless mixed-reality experience.

A continuous data feed that is tailored for mobile bandwidth limitations is used to control AR streaming. To guarantee that the AR view stays fluid even in the face of fluctuating network conditions, strategies like adaptive streaming and data compression are used. Here, Normcore and ARCore integration is essential because it enables the quick delivery of interaction and visual data to mobile devices.

Additionally, the overlay mechanism includes features to adapt to variations in display characteristics and device orientation. This flexibility guarantees that the user will always see a consistent and engaging view of the VR environment, irrespective of the AR device being used. This process is made easier by the use of standardized APIs and protocols, which give the system the strength and adaptability to handle a variety of hardware configurations.

3.4.3 AR Environment Adaptation

The system incorporates mechanisms for adjusting the virtual environment to the limitations of mobile devices in order to give AR users the best possible experience. This entails dynamically modifying the object details and rendering quality in accordance with the device's performance capabilities. The system guarantees that AR viewers have a top-notch experience without taxing their hardware in the process.

The feedback loop created by Normcore's synchronization system is intimately related to the adaptation process. The AR module evaluates the network's current condition while data is being transmitted in real time and modifies the level of detail as necessary. Even in situations with high user loads, this adaptive approach helps to maintain low latency and guarantees that the AR view stays responsive and captivating.

Additionally, integrating user interface components made especially for mobile screens is part of the adaptation of the AR environment. These user interface components give

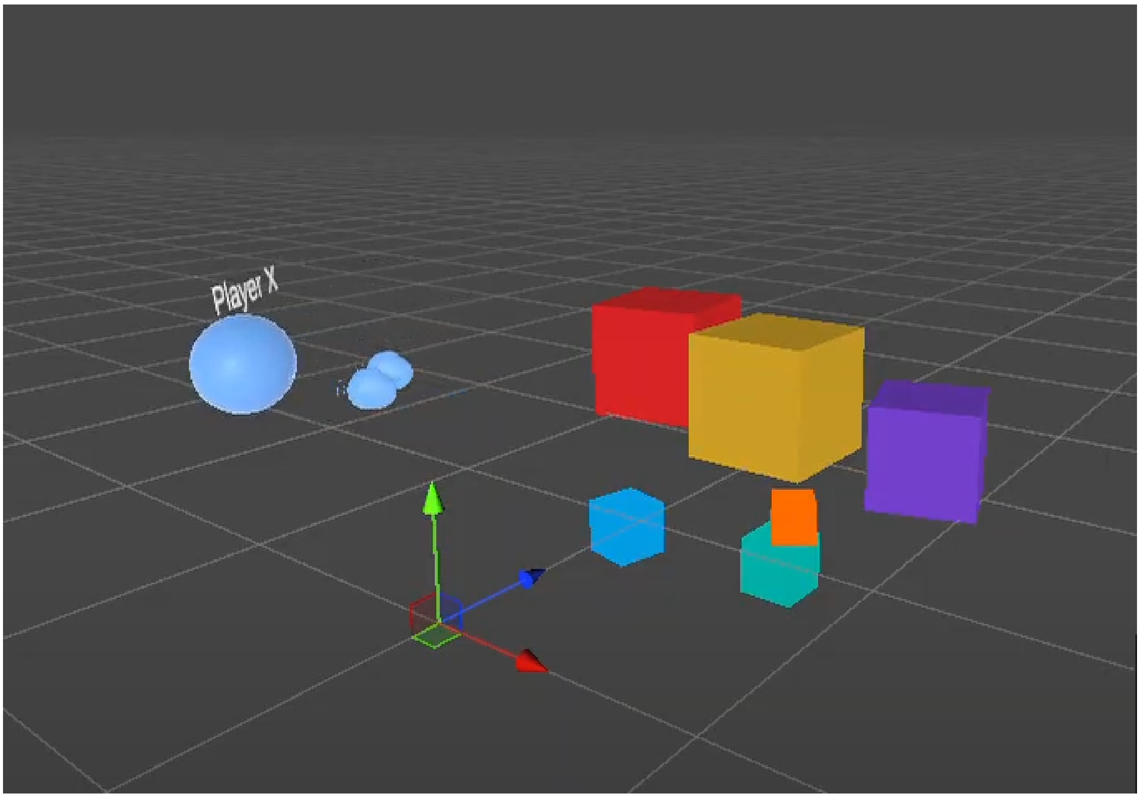

FIGURE 3.5: AR Spectator

AR users more context and control options, enabling them to engage meaningfully with the VR environment. The foundation of the AR experience on the platform is made up of adaptive rendering and user-friendly UI design.

3.5 Avatar Customization and Room Management

The procedures for managing rooms and customizing avatars are explained in detail in this section. In order to create an immersive and unique experience, it describes how users can alter their avatars and control room settings, including environment selection and dynamic updates. The significance of maintaining a consistent user identity is emphasized by the discussion of synchronizing these customizations across all clients.

3.5.1 Avatar Customization Process

Avatar customization, which enables users to alter their look before entering the immersive environment, is one of the system's primary features. The process begins in the pregame lobby, where users input their names and select their preferred avatar colors and features. Personalization not only boosts user engagement but also makes it easier for players to create unique personas in the multiplayer setting.

The customization process is meant to be straightforward and user-friendly. The system provides real-time previews of avatar changes, utilizing Unity's UI capabilities to give users immediate feedback. Normcore is essential in ensuring that each user's avatar is shown consistently throughout the session by ensuring that these changes are synchronized across all clients.

Additionally, there is room to expand the avatar customization process. Later iterations of the system may include more complex customization options, such as clothing, accessories, and animations. The platform's versatility allows it to be utilized in a range of contexts, including formal educational settings and dynamic product showrooms.

3.5.2 Room Creation and Management

Since it determines how users join and interact in the shared environment, room management is an essential part of the platform. Users can join already-existing rooms or create new ones using the system; the person who creates the room gets extra rights to alter the setting. Choosing themes (such as sci-fi, nature, or modern) and establishing guidelines for interactions within the space are a couple examples of these customizations.

To guarantee that users can quickly set up a session without requiring a lot of configuration, the room creation process has been simplified. In order to guarantee that every user is correctly registered and that the virtual environment is uniform across all devices, Normcore manages room authentication and session management. This method lessens the complexity that comes with establishing multiplayer sessions.

Figure 3.6: AR Spectator

An additional crucial element of efficient room management is dynamic control during sessions. Room creators can alter the setting at any time, and everyone involved will notice the changes right away. This real-time adaptability is crucial for maintaining an engaging and responsive experience in scenarios where the room's context (like a training session or product demonstration) may change over time.

3.5.3 Mirror Self-Visualization Feature

One of the platform's unique features is the mirror self-visualization function, which enables users to view their avatars in real time. This feature is essential for ensuring that user presence is reinforced and that customizations are appropriately reflected. Carefully placed mirrors in the virtual room provide a continuous self-view, allowing users to check how their avatar looks and make any necessary adjustments.

The mirror feature is seamlessly integrated with Normcore's synchronization system. Users' avatar modifications are immediately reflected in the mirrored view and on all other connected clients. Because of this continuous visual feedback, users in the VR/AR environment feel more engrossed and secure in their individual identities.

FIGURE 3.7: Mirror Self Visualization Feature

Additionally, the mirror self-visualization feature is a helpful tool for user engagement and interface validation. By allowing both inexperienced and experienced users to get to know the virtual environment, it strengthens the system's emphasis on intuitive and user-centric design.

3.6 Voice Communication and Interaction

This section looks at how voice communication has been incorporated into the platform to enable low-latency, real-time user interactions. The use of feedback systems and audio streaming protocols that complement visual and interaction cues is covered. Maintaining the collaborative aspect of the immersive environment requires clear, instantaneous voice communication.

3.6.1 Voice Chat Integration

A key component of real-time collaboration in immersive settings is voice communication. Voice chat features are integrated into the system to enable smooth user communication. The platform makes sure that voice interactions are timely and clear by using low-latency transmission protocols and optimized audio codecs, which improves the immersive experience overall.

The voice chat integration is made to blend in seamlessly with the AR and VR user interfaces. In spite of the device, users can start and join voice conversations, guaranteeing that technical constraints won't impede cooperative tasks and interactions. The real-time networking features of Normcore aid in keeping audio streams synchronized, ensuring that every spoken word is transmitted across the network in a timely manner.

To reduce latency and jitter in voice communication, extensive testing and optimization have been done. The system makes use of error-correction protocols and adaptive buffering techniques to guarantee steady audio quality even in settings with fluctuating network conditions. Supporting cooperative situations like interactive product demos and training sessions requires this strong voice chat integration.

3.6.2 Low-latency Audio Transmission

Low-latency audio transmission is essential for preserving the organic conversational flow in immersive settings. The system uses cutting-edge audio streaming protocols that prioritize and compress voice data to lower latency. This method maintains the real-time aspect of the interaction by guaranteeing that users encounter little delay.

To further improve audio performance, specialized hardware and software optimizations are used. The platform is made to function well on a variety of gadgets, including mobile AR devices and expensive VR headsets. These adjustments are necessary to preserve voice communications' fidelity and immediacy, which are critical for productive user engagement and dynamic teamwork.

The close integration of Normcore's networking layer with the system's audio transmission module ensures that voice packets are effectively prioritized and managed alongside other important data. Voice communication is still a dependable and efficient means of collaboration thanks to this integrated approach, which enables smooth interaction between the immersive experience's audio and visual elements.

3.6.3 Interaction Feedback and Notifications

To improve user engagement, the system offers real-time interaction feedback and notifications in addition to voice chat. This comprises haptic feedback, visual cues, and on-screen notifications that alert users to important occurrences or environmental changes. These kinds of feedback systems are crucial for maintaining user awareness and interest, particularly during intricate exchanges.

Both VR and AR platforms are compatible with the feedback system. For instance, a visual confirmation appears on AR devices as well as in the VR environment when a user interacts with an object. The immersive experience is strengthened by this cross-platform feedback, which makes sure that every user stays in sync with the ongoing interactions.

Additionally, the synchronization and session management layers are integrated with the notification system. All connected clients receive real-time updates, such as changes to the environment or new user entries. This further improves teamwork and clears up confusion during dynamic interactions by guaranteeing that all participants are promptly informed of significant events.

3.7 Testing and Simulation Environment

The testing frameworks and simulation tools used to confirm the platform's dependability and performance are described in this section. It describes how local and cross-platform testing is carried out using tools like ParrelSync and XR Simulation to make sure the system satisfies strict performance requirements. Additionally emphasized is the significance of thorough testing in locating bottlenecks and enhancing functionality.

3.7.1 Multiplayer Testing with ParrelSync

Extensive testing is a crucial step in the development process. The system uses a key tool, ParrelSync, for local multiplayer testing. ParrelSync allows developers to thoroughly test real-time synchronization and interaction mechanics in a controlled environment by simulating multiple game instances on a single computer.

This testing phase's objective is to verify that all of the system's features—such as real-time object synchronization and avatar customization—function as planned under various conditions. By simulating multiple users, developers can identify potential bottlenecks and latency issues that could affect performance. Before deployment, the system is modified based on the insights gained from these tests to ensure that it meets the required performance standards.

FIGURE 3.8: VR Keyboard

Additionally, iterative development is made possible by the use of ParrelSync. Continuous testing makes sure that modifications don't impair the system's overall functionality when new features are added or existing components are optimized. For a multiplayer environment to remain responsive and stable, this iterative process is essential.

3.7.2 XR Simulation on Laptop Devices

In addition to local multiplayer testing, the system undergoes XR simulation on laptops to ensure cross-platform compatibility. This testing phase ensures that the platform's key features function consistently across a variety of devices by mimicking the user experience on less specialized hardware. The behavior of VR and AR components is replicated using XR simulation tools in a controlled, laptop-based environment.

Developers can test the integration of Normcore and ARCore functionalities in the XR simulation environment without the need for costly VR headsets. This broadens the testing scope and aids in identifying performance issues that could arise on different hardware configurations. The simulation results inform the necessary optimizations, particularly in the areas of data synchronization and rendering quality, to guarantee a smooth user experience.

Validating scalability also requires extensive simulation testing. Developers can evaluate the system's performance in various scenarios by modeling different user loads. In order to prepare for a scalable worldwide deployment, the knowledge gained from these simulations is applied to improve the networking protocols as well as the data management techniques.

3.7.3 Performance and Scalability Testing

Testing for performance and scalability is done to make sure the system can support many users at once without sacrificing functionality or responsiveness. Key metrics like latency, data throughput, and synchronization accuracy are measured during this phase, which also entails stress testing the networking infrastructure. The main objective is to determine the maximum user capacity and optimize the system for maximum efficiency.

The system's resilience is assessed using a variety of testing scenarios, including both simulated and real-world circumstances. Especially when there is a lot of traffic, these tests aid in identifying possible communication channel bottlenecks. By optimizing Normcore's protocols based on the data gathered during these tests, the platform is guaranteed to maintain low latency and high reliability even when heavily used.

The integration of AR streaming is another area of focus for scalability testing. Developers can make sure that AR viewers have a consistent and engaging experience by testing Google ARCore's performance in various network scenarios. The thorough performance testing process ensures that the system is reliable and prepared for deployment in a variety of real-world situations.

3.8 Data Management and Logging

The methods for handling session data and recording in-the-moment interactions on the platform are explained in this section. In order to support ongoing development and troubleshooting, it focuses on the storage and analysis of user data, synchronization events, and performance metrics. Enabling thorough performance analysis and preserving system integrity require strong data management procedures.

3.8.1 Session Data Storage and Retrieval

Effective data management is essential to the immersive experience's continuity and integrity. Session data, such as environment settings, user profiles, and avatar customizations, are captured by a robust data storage mechanism integrated into the system. Session information can be accurately and quickly retrieved because this data is stored in a single location.

The architecture of the data storage system supports both real-time retrieval and historical analysis. Throughout a session, all clients' live data is updated and synchronized continuously using Normcore's data propagation protocols. Simultaneously, session logs are stored for post-session analysis, which aids developers in monitoring user interactions, monitoring performance, and identifying potential issues.

Secure data management procedures are also used to guarantee data integrity and user privacy. The system incorporates encryption and access control measures to protect private data from unwanted access. Both the platform's operational success and adherence to data protection laws depend on this thorough data management strategy.

3.8.2 Analytics and User Behavior Tracking

The system's data management strategy must include analytics and user behavior monitoring. By monitoring user interactions, such as communication patterns, object manipulations, and avatar movements, the platform collects valuable data that can be used to improve the user experience. These analytics provide information about how users engage with the system by highlighting areas that may require further optimization.

The collected data is processed in real time before being stored for future research. Developers can use this data to understand usage patterns, identify trends, and evaluate the effectiveness of a specific feature. For example, keeping track of how frequently users change their avatars or interact with specific items can help prioritize features and direct future enhancements.

Additionally, scalability testing heavily relies on analytics. Developers can evaluate the overall effect of increased load on the platform by tracking user behavior data and system performance metrics. By taking a comprehensive approach to data analytics, the system is kept user-centric and responsive, allowing for ongoing development.

3.8.3 Data Synchronization Logs

An important record of how the platform manages real-time updates is provided by data synchronization logs. These logs document the time stamps of data transmission between clients and capture all interactions, including moving avatars and modifying objects. They guarantee that any synchronization issues can be promptly found and fixed, which makes them essential for debugging and performance analysis.

Normcore's networking layer continuously monitors data flow between the VR and AR modules, keeping the synchronization logs current. In addition to guaranteeing that every action is precisely duplicated across devices, this logging mechanism offers a historical record for future use. These logs can be examined by developers to improve synchronization protocols and reduce latency.

The system facilitates thorough testing and evaluation and guarantees transparency in data management through the integration of detailed logging. These logs are essential for verifying the synchronization algorithms' effectiveness and making sure the immersive experience is consistent across a range of scenarios.

3.9 Future Enhancements and Methodological Refinements

An outlook on future system improvements and methodological improvements is provided in this section. It describes potential future paths like continuous user feedback loops, AI-driven adaptation, and integration with new XR devices. With these improvements, the platform will continue to develop and stay state-of-the-art while adapting to new technological developments in immersive settings.

3.9.1 Potential Integration of AI-driven Adaptation

One promising approach to future platform improvement is the addition of AI-driven adaptation mechanisms. The system could respond to user behavior and current network conditions by dynamically adjusting synchronization settings, rendering quality, and user interface elements using machine learning algorithms. Even in the event of fluctuating user load, this adaptable approach would help maintain optimal performance.

Additionally, the application of AI may lead to more personalized experiences. For instance, predictive analytics may optimize object synchronization according to usage trends, and natural language processing and adaptive voice recognition may improve the efficacy of communication. These enhancements would significantly increase the system's overall scalability and responsiveness, ensuring that it remains robust as technology develops.

Future research will focus on examining these AI integrations, developing prototypes, and assessing their performance in real-world scenarios. The goal is to create a self-optimizing system that continuously learns from user interactions and network conditions to enable more responsive and intuitive XR experiences.

3.9.2 Future Device and Technology Incorporation

The platform must change as new XR gadgets and technologies appear in order to smoothly integrate these developments. Expanding compatibility to include next-generation VR headsets, AR glasses, and other immersive devices will be the main goal of future system iterations. This proactive strategy guarantees that, despite hardware variations, the platform will continue to be relevant and able to provide high-quality experiences.

In order to accommodate a wider variety of input modalities and display technologies, integrating additional technologies will require improving the current architecture and updating synchronization protocols. The platform's modular architecture allows for easy adaptation to these changes. Improved sensor integration for more accurate tracking and engaging user interactions might also be included in future updates.

Continued evolution is necessary to stay ahead in the rapidly evolving field of XR. New devices and technologies will be added to the system, expanding its user base and opening up new application areas, further demonstrating its adaptability and scalability.

3.9.3 Continuous Improvement and User Feedback Loop

A strong user feedback loop for ongoing improvement is a crucial part of the development strategy for the future. Frequent gathering and evaluation of user input will direct iterative system improvements, guaranteeing that the platform adapts to real user requirements and emerging technology. Direct observations made during test sessions, usage analytics, and surveys will all be a part of this feedback loop.

User input is crucial for spotting unanticipated problems and confirming the usefulness of new features. Developers can prioritize improvements that have the biggest effects on system performance and user satisfaction by interacting with the user community. A key component of the platform's long-term viability and relevance is its user-centric strategy.

Furthermore, all facets of the system—from voice communication and network synchronization to avatar customization and AR streaming—will be subject to continuous improvement practices. The dedication to continuous improvement guarantees that the platform will continue to be state-of-the-art, flexible, and able to offer both VR and AR users an exceptional immersive experience.

Chapter 4

Results and Discussions

Objective

Creating a comprehensive VR/AR multiplayer platform that facilitates smooth, real-time user interaction across multiple devices and environments is the main goal of this project. The platform's goal is to create immersive experiences where users can interact with digital content, collaborate, and communicate in both shared virtual spaces and augmented physical settings by combining virtual reality (VR) and augmented reality (AR) technologies. In order to provide a seamless and engaging user experience, this entails putting in place reliable networking solutions that guarantee low-latency data transfer and precise synchronization of user actions and environmental changes.

Designing and implementing a scalable architecture that supports a variety of applications, such as interactive gaming, virtual meetings, and educational tools, is another important goal. The platform aims to improve user engagement and personalization by offering customizable avatars and user-friendly interfaces. In order to guarantee accessibility and adaptability, the project also intends to integrate cutting-edge features like voice communication, real-time data logging, and cross-platform compatibility. By accomplishing these goals, the project hopes to advance the state of VR/AR technology and lay the groundwork for further advancements in immersive, team-based technologies.

4.1 Results and Discussions

This section presents the performance analysis and experimental results of the proposed VR/AR multiplayer system. By focusing on crucial metrics like network latency, object synchronization accuracy, and real-time update consistency, the results compare traditional approaches with the Normcore-based approach. Showing how Normcore's integration with Google ARCore and Unity significantly improves the user experience in a range of use cases, such as product showrooms, classrooms, and meditation settings, requires these results.

4.2 Results and Comparison

The performance analysis and experimental findings of the suggested VR/AR multi-player system are shown in this section. The results contrast conventional methods with the Normcore-based approach by concentrating on important metrics such as network latency, object synchronization accuracy, and real-time update consistency. These outcomes are necessary to demonstrate how Normcore's integration with Google AR-Core and Unity greatly enhances the user experience in a variety of use cases, including product showrooms, classrooms, and meditation settings.

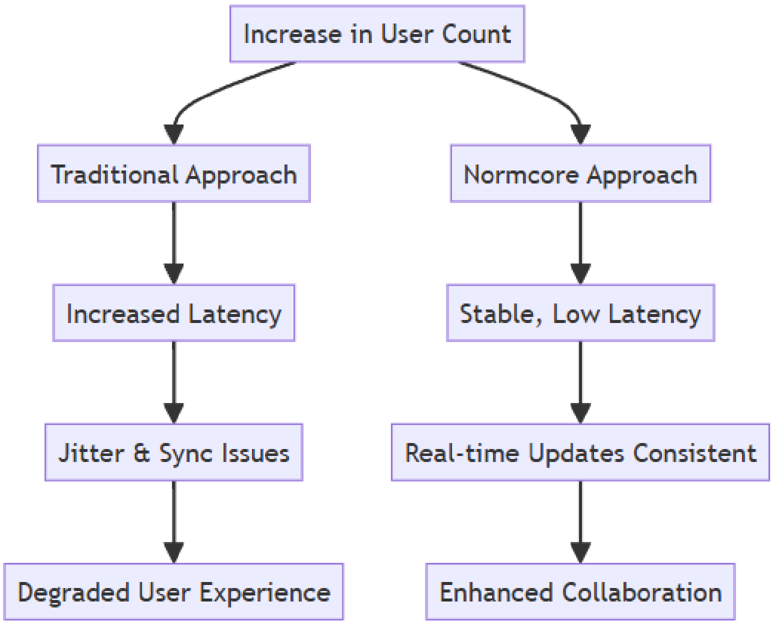

FIGURE 4.1: Performance Comparison Flowchart

4.2.1 Network Latency vs. User Count – Traditional Approach

Network latency was measured in experiments with conventional networking solutions as the number of concurrent users rose. The findings showed that the average latency increased dramatically with the number of users joining the session, frequently surpassing acceptable thresholds for real-time interactions. According to the data, latency spikes were frequent and caused observable lags when manipulating objects and speaking.

Subsequent investigation showed that inconsistent experiences were caused by the latency variability, especially in situations with a high user density. The conventional method had trouble staying in sync, which had a negative effect on user involvement. It was challenging to maintain a large-scale, immersive multiplayer environment with reliable performance because of these constraints.

All things considered, situations requiring high responsiveness and low latency were not well served by conventional networking techniques. The need for a more reliable solution like Normcore was prompted by the increased delays, which not only broke the interactive flow but also diminished the overall immersion.

4.2.2 Object Synchronization Accuracy vs. Scale – Traditional Approach

Another crucial metric that was assessed using conventional techniques was object synchronization accuracy. Significant differences in the shared virtual environment were observed in these experiments, even when there were only minor delays in updating object positions across devices. Jitter and misaligned object states across clients were the result of synchronization errors that worsened as the number of interacting objects rose.

According to the study, these errors were especially troublesome in applications that needed accurate object manipulation, like training simulations and virtual product demos. The discrepancies made it more difficult to work together and made the surroundings seem less realistic. When the virtual objects were inconsistently updated, users reported feeling confused.

Therefore, the inability of the conventional method to maintain accurate object synchronization brought attention to the need for a better solution that could scale effectively and manage high-frequency updates. A major contributing factor to the move toward Normcore-based synchronization was this shortcoming.

4.2.3 Network Latency vs. User Count – Normcore Approach

The system's network latency performance significantly improved when Normcore was used as the networking solution. The tests demonstrated that even when the number of concurrent users rose, Normcore was able to maintain low and consistent latency levels.

The built-in templates and optimized data replication protocols of Normcore, which effectively handle real-time communication, are responsible for this consistency.

In-depth tests showed that the latency stayed within reasonable bounds, guaranteeing that user actions—like moving the avatar or interacting with objects—were nearly instantly reflected on all devices. This enhancement was essential to preserving a smooth and engaging VR/AR experience, especially in high-interaction situations like virtual classrooms and product showrooms.

In addition to improving the system's real-time responsiveness, Normcore's notable latency reduction made for a more seamless and interesting user experience. These outcomes highlight the benefits of utilizing a customized networking solution designed for immersive applications.

4.2.4 Real-time Update Consistency – Comparison between Traditional and Normcore Approaches

The Normcore technique significantly improves real-time update consistency when compared directly to conventional techniques. The conventional method resulted in a fragmented and disjointed user experience due to frequent object state discrepancies and delayed voice communications. As opposed to this, Normcore's architecture made sure that all updates, including those pertaining to avatar movements and environmental changes, were precisely and quickly synchronized across all linked devices.

With little jitter and almost instantaneous change propagation, the Normcore-based system showed a high degree of consistency. In intricate interactive sessions, where even slight delays could impede the flow of collaboration, this dependability was especially noticeable. In addition to making VR users more immersed, the increased update consistency made sure that viewers of AR viewed the virtual world in a consistent and cohesive manner.

Overall, the comparative analysis demonstrates that the Normcore approach provides a strong remedy for the problems that conventional networking techniques encounter. A better user experience is directly correlated with improved real-time synchronization performance, which makes the platform suitable for widespread implementation in a variety of applications.

Chapter 5

Conclusion and Future Scope

The objective of developing an immersive environment where users can interact seamlessly across virtual and augmented realities has been accomplished with the development of the VR/AR multiplayer platform. The platform provides a seamless and captivating user experience by incorporating cutting-edge networking solutions and guaranteeing real-time synchronization. This achievement highlights how VR and AR technologies have the potential to transform teamwork across a range of fields.

5.1 Summary of Achievements

A VR/AR multiplayer platform that facilitates smooth, real-time user interaction across various devices and environments was successfully developed by the project. A seamless and engaging user experience was made possible by our integration of reliable networking solutions, which guaranteed low-latency data transfer and precise user action synchronization. User engagement and personalization were further improved by the use of adaptable avatars and user-friendly interfaces.

Furthermore, a wide range of applications, such as interactive gaming, virtual meetings, and educational tools, are supported by the platform's scalable architecture. The addition of voice communication and real-time data logging capabilities made the system more complete and adaptable. Together, these accomplishments show how the platform has the ability to completely transform immersive teamwork.

5.2 Challenges Encountered

A number of difficulties arose during the development process. Since differences in device capabilities affected the consistency of the user experience, one major challenge was guaranteeing consistent performance across various hardware configurations. Careful design considerations were also necessary to address users' physical discomfort and motion sickness in order to reduce negative effects during immersive sessions.

Preserving user privacy and data security in the multiplayer setting presented another difficulty. To safeguard sensitive data and preserve user confidence, strong encryption and secure data handling procedures had to be put in place. These difficulties made it clear that system design and implementation require constant optimization and attention to detail.

5.3 Integration of Emerging Technologies

In the future, incorporating cutting-edge technologies like machine learning and artificial intelligence (AI) can greatly expand the platform's functionality. The immersive experience can be enhanced by using AI to produce non-player characters (NPCs) that are more adaptive and realistic. By predicting user preferences and personalizing interactions based on user behavior, machine learning algorithms can increase user satisfaction and engagement.

Furthermore, more natural and intuitive interactions within the virtual environment can be achieved by integrating developments in eye-tracking and haptic feedback technologies. These integrations could improve the platform's responsiveness and realism, giving users a more engaging and immersive experience.

5.4 Expansion into New Applications

The platform's adaptability creates chances for growth into other fields outside of the original applications. The platform can be modified for patient education, medical training simulations, and virtual therapy sessions in the healthcare industry, offering secure and regulated learning and treatment environments. Through the creation of interactive shopping spaces and virtual product trials, augmented reality features in retail can improve the customer experience.

The platform can also be used in the tourism sector to provide virtual tours of historical locations and cultural landmarks, opening them up to a worldwide audience. These extensions show how the platform's innovative and immersive solutions have the potential to influence a variety of industries.

5.5 Addressing Ethical and Social Implications

The ethical and social ramifications of VR and AR technologies must be addressed as they become more prevalent in daily life. To preserve trust and protect user rights, it is critical to guarantee user privacy, data security, and informed consent. Developers must give users control over their personal data and adopt transparent data practices.

In order to serve users with a range of backgrounds and abilities, the platform must also encourage accessibility and inclusivity. An environment that is both equitable and easy to use can be achieved by designing features that are flexible and take into account different user needs.

5.6 Future Research Directions

In order to accommodate bigger user bases and more intricate interactions, future research should concentrate on improving the platform's performance and scalability. To do this, it will be essential to look into ways to lower latency and increase synchronization accuracy. Furthermore, investigating how cloud computing and 5G technology can work together can make experiences more accessible and responsive.

Research into mitigating motion sickness and improving user comfort during extended use is also necessary. Developing adaptive systems that can adjust to individual user sensitivities and preferences will enhance the overall experience. Furthermore, studying the long-term psychological and social effects of immersive environments can provide valuable insights into their impact and inform responsible development practices.

In conclusion, a strong basis for immersive collaborative technologies has been established by the VR/AR multiplayer platform project. The platform can adapt to the increasing needs of diverse industries by tackling present issues and seizing new opportunities, providing creative answers and improving user experiences.

Chapter 6

Appendix

This appendix provides supplementary materials relevant to the development of the VR/AR multiplayer platform, including sample code snippets, configuration settings, and additional resources.

6.1 Sample Code Snippets

6.1.1 Player Connection Handling

Below is a sample script to Spawns a dummy avatar as a child of the XR Origin's Camera Offset at the start of the scene:

Listing 6.1: Avatar Spawner in Unity

```
using UnityEngine;

public class AvatarSpawner : MonoBehaviour
{
    public GameObject dummyAvatarPrefab;
    public Transform xrOriginParent;
    // Drag XR Origin's Camera Offset here

    void Start()
    {
        GameObject avatar =
        Instantiate(dummyAvatarPrefab, xrOriginParent);
        avatar.transform.localPosition = Vector3.zero;
```

```
        avatar.transform.localRotation = Quaternion.identity;
    }
}
```

6.1.2 Object Synchronization

The following script demonstrates synchronizing object transformations across the network:

Listing 6.2: AObject Synchronization

```
using Unity.Netcode;
using UnityEngine;

public class NetworkObjectSync : NetworkBehaviour
{
    private NetworkVariable<Vector3> objectPosition =
    new NetworkVariable<Vector3>();

    private void Update()
    {
        if (IsOwner)
        {
            objectPosition.Value = transform.position;
        }
        else
        {
            transform.position = objectPosition.Value;
        }
    }
}
```

6.1.3 Name Tag Updater

Listing 6.3: Name Tag Updater

```
using TMPro;
using UnityEngine;

public class NameTagUpdater : MonoBehaviour
```

```
{
    public TextMeshProUGUI nameTag;
    public Transform headTransform;
    public Vector3 offset = new Vector3(0, 0.2f, 0);

    void Update()
    {
        if (nameTag == null || headTransform == null ||
        Camera.main == null)
        {
            Debug.LogError("NameTagUpdater is missing a reference!");
            return;
        }
        nameTag.text = CustomizationData.PlayerName;
        nameTag.transform.position = headTransform.position + offset;
        nameTag.transform.rotation =
        Quaternion.LookRotation(
            nameTag.transform.position - Camera.main.transform.position);
    }
}
```

6.1.4 Networked Avatar

Listing 6.4: Networked Avatar

```
using UnityEngine;
using Normal.Realtime;
using Normal.Realtime.Serialization;
using TMPro;

[RealtimeModel]
public partial class AvatarModel
{
    [RealtimeProperty(1, true, true)] private string _playerName;
    [RealtimeProperty(2, true, true)] private int _colorIndex;
}

[RequireComponent(typeof(RealtimeView), typeof(RealtimeTransform))]
public class NetworkedAvatar : RealtimeComponent<AvatarModel>
```

```
{
    public TextMeshProUGUI nameTag;
    public MeshRenderer[] bodyMeshes;
    [SerializeField] private Material[] colorMaterials;
    // Assign materials in Inspector (Red, Blue, Green, Yellow, Cyan)

    private void Start()
    {
        if (realtimeView.isOwnedLocallySelf)
        {
            model.playerName =
            PlayerPrefs.GetString("PlayerName", "Player");
            model.colorIndex = PlayerPrefs.GetInt("ColorIndex", 0);
        }

        UpdateAvatarAppearance();
    }

    protected override void OnRealtimeModelReplaced
    (AvatarModel previousModel, AvatarModel currentModel)
    {
        if (currentModel != null)
        {
            UpdateAvatarAppearance();
        }
    }

    private void Update()
    {
        if (model != null)
        {
            UpdateAvatarAppearance();
        }
    }

    private void UpdateAvatarAppearance()
    {
        if (nameTag != null && model != null)
        {
```

```
            nameTag.text = model.playerName;
        }

        if (bodyMeshes != null && model != null &&
        colorMaterials != null && colorMaterials.Length > 0)
        {
            int index =
            Mathf.Clamp(model.colorIndex, 0, colorMaterials.Length - 1);
            Material mat = colorMaterials[index];
            foreach (var mesh in bodyMeshes)
            {
                if (mesh != null)
                {
                    mesh.material = mat;
                }
            }
        }
    }
}
```

6.1.5 Pregame Customization

Listing 6.5: Pregame Customization

```
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;
using TMPro;
using Normal.Realtime;
using System.Collections.Generic;

public class PreGameCustomization : MonoBehaviour
{
    public TMP_InputField nameInputField;
    public TMP_Dropdown colorDropdown;
    public TMP_InputField roomInputField;
    public Button createRoomButton;
    public Button joinRoomButton;
    public Material[] colorMaterials; // Assign 5 materials in Inspector (
```

```
void Start()
{
    if (!ValidateUIComponents()) return;

    InitializeColorDropdown();
    nameInputField.onValueChanged.AddListener(UpdatePlayerName);
    colorDropdown.onValueChanged.AddListener(UpdatePlayerColor);
    createRoomButton.onClick.AddListener(OnCreateRoomClicked);
    joinRoomButton.onClick.AddListener(OnJoinRoomClicked);
    Debug.Log("PreGameCustomization initialized");
}

bool ValidateUIComponents()
{
    if (nameInputField == null)
    Debug.LogError("Name Input Field is not assigned!");
    if (colorDropdown == null)
    Debug.LogError("Color Dropdown is not assigned!");
    if (roomInputField == null)
    Debug.LogError("Room Input Field is not assigned!");
    if (createRoomButton == null)
    Debug.LogError("Create Room Button is not assigned!");
    if (joinRoomButton == null)
    Debug.LogError("Join Room Button is not assigned!");
    if (colorMaterials == null ||
    colorMaterials.Length < 5)
    Debug.LogError("Color Materials array is not properly assigned!");
    return nameInputField != null &&
    colorDropdown != null &&
    roomInputField != null &&
            createRoomButton != null && joinRoomButton != null
            && colorMaterials != null && colorMaterials.Length >= 5;
}

void InitializeColorDropdown()
{
    colorDropdown.ClearOptions();
    colorDropdown.AddOptions(new List<string>
```

```
        { "Red", "Blue", "Green", "Yellow", "Cyan" });
    }

    void UpdatePlayerName(string newName)
    {
        CustomizationData.PlayerName
        = string.IsNullOrEmpty(newName) ? "Player" : newName;
    }

    void UpdatePlayerColor(int colorIndex)
    {
        CustomizationData.ColorIndex = colorIndex;
        CustomizationData.PlayerColor = colorMaterials[colorIndex].color;
    }

    void OnCreateRoomClicked()
    {
        Debug.Log("Create button clicked");
        SaveCustomizationData();
        string roomName =
        string.IsNullOrEmpty(roomInputField.text) ? "DefaultRoom" : roomIn
        PlayerPrefs.SetString("RoomName", roomName);
        PlayerPrefs.SetInt("IsCreator", 1);
        Debug.Log($"Creating room: {roomName}");
        SceneManager.LoadScene("1_CommonRoom");
    }

    void OnJoinRoomClicked()
    {
        Debug.Log("Join button clicked");
        SaveCustomizationData();
        string roomName = string.IsNullOrEmpty(roomInputField.text) ? "Defa
        PlayerPrefs.SetString("RoomName", roomName);
        PlayerPrefs.SetInt("IsCreator", 0);
        Debug.Log($"Joining room: {roomName}");
        SceneManager.LoadScene("1_CommonRoom");
    }

    void SaveCustomizationData()
```

```
    {
        PlayerPrefs.SetString("PlayerName", CustomizationData.PlayerName);
        PlayerPrefs.SetInt("ColorIndex", CustomizationData.ColorIndex);
    }
}
```

6.2 Configuration Settings

6.2.1 Unity Project Settings

For optimal performance in VR/AR applications, the following Unity project settings are recommended:

- **Scripting Backend**: IL2CPP
- **API Compatibility Level**: .NET Standard 2.1
- **Target Architectures**: ARM64
- **Graphics API**: Vulkan (for Android) or Direct3D11 (for Windows)

Source: Unity's VR Multiplayer Template Quick Start Guide.

6.2.2 XR Plugin Management

Ensure that the appropriate XR plugins are enabled:

- **OpenXR**: Enabled for cross-platform VR/AR support
- **Oculus XR Plugin**: Enabled if targeting Oculus devices

Source: Unity's VR Multiplayer Template Quick Start Guide.

6.3 Additional Resources

- **Unity VR Multiplayer Template**: A comprehensive starting point for developing VR multiplayer applications.

- **AR Foundation Samples**: A collection of AR Foundation sample projects demonstrating various features and use cases.
- **Oculus Samples**: A repository of sample projects provided by Oculus to demonstrate VR development techniques.

This appendix serves as a reference for developers seeking to understand and implement key components of the VR/AR multiplayer platform.

Bibliography

[1] D. Lee, A. K. Sanyal, E. A. Butcher, and D. J. Scheeres, *"Finite-time control for spacecraft body-fixed hovering over an asteroid,"* IEEE Transactions on Aerospace and Electronic Systems, vol. 51, no. 1, pp. 506–520, Jan. 2015, doi: 10.1109/TAES.2014.140197.

[2] S. Keränen, *"Content Management: Concept and Indexing Term Equivalence in a Multilingual Thesaurus,"* Pantelidis, V. S., [(2009].

[3] Jason Jerald. *The VR Book: Human-Centered Design for Virtual Reality.* Association for Computing Machinery, 2015.

[4] C. R. S. Maharani and S. Das, "Cross-Platform AR/VR Development: Strategies for Seamless User Experiences," *An Online Peer Reviewed / Refereed Journal*, vol. 2, no. 9, pp. 255, Sep. 2024.

[5] G. Di Gironimo and A. Lanzotti, "Designing in VR," *Int. J. Interact. Des. Manuf.*, vol. 3, pp. 51–53, 2009. doi:

[6] I. R. Mukhametkhanov, M. R. Khafizov, and A. V. Shubin, "Comparison of client-server solutions in the development of multiplayer online games on Unity," *Electronic Libraries*, vol. 25, no. 5, pp. 472–488, Dec. 2022. doi:

[7] J. Smed, "Networking for computer games," *Int. J. Comput. Games Technol.*, vol. 2008, Art. ID 928712, p. 1, 2008. doi: 10.1155/2008/928712.

[8] Y. Liu and Y. Li, "Virtual reality system for industrial training," in *Proc. Int. Conf. Virtual Reality and Visualization (ICVRV)*, Qinhuangdao, China, 2020, pp. 338-339. doi:

[9] P. Pareek, "Designing Multiplayer Experiences in AR and VR: Challenges and Opportunities," *ShodhKosh: Journal of Visual and Performing Arts*, vol. 5, no. ICETDA24, pp. 360–366, May 2024. doi:

[10] U. Žákovská and D. Sedláček, "Studying the Role of Non-Player Users (Spectator and Operator) in XR Single and Multiplayer Experiences," in *Proceedings

of the 2023 14th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, Sep. 2023, pp. 129–134. doi:

[11] B. Hu, Y. Zhang, S. Hao, and Y. E. Tao, "MOFA: Exploring Asymmetric Mixed Reality Design Strategy for Co-located Multiplayer Between Handheld and Head-mounted Augmented Reality," *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023.

[12] A. Stavroulakis, D. Dimelli, M. Roumeliotis, and A. Mania, "An Augmented Reality System Architecture for Flood Management," in *Proc. Int. Conf. Geogr. Inf. Syst. Theory, Appl. Manage.*, 2024.

[13] T. Fu and H. Zhang, "The System Architecture and Decision Mechanism of the Deterministic Application Layer in the Industrial Internet," *Proc. 3rd Int. Conf. Consum. Electron. Comput. Eng. (ICCECE)*, 2023, pp. 18–25.

[14] Peaker Map, "Cross-Platform VR/AR Development: Build Once, Deploy Everywhere," December 21, 2024.

[15] "Strategies for Cross-Platform Web AR Development," Aircada™, Inc., Aug. 25, 2023. [Online].

[16] S. Frish, M. Druchok, and H. Shchur, *"Molecular MR Multiplayer: A Cross-Platform Collaborative Interactive Game for Scientists,"* Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology (VRST '20), Association for Computing Machinery, 2020. [Online].
Dept. of Computer Science and Engineering, St. Joseph's College of Engineering and Technology, Palai, Kerala, India : International Conference on Soft-Computing and Network Security (ICSNS), 2015.