TECHNISCHE UNIVERSITÄT ILMENAU

MEDIA PROJECT REPORT

Meet and Greet MR

AUTHOR:
SAMEER KAZMI, TAIMOOR AHMED, UMAR IRSHAD

A MEDIA PROJECT REPORT SUBMITTED
IN THE

VIRTUAL WORLDS AND DIGITAL GAMES GROUP
INSTITUTE OF MEDIA TECHNOLOGY

SEPTEMBER 30, 2021

TECHNISCHE UNIVERSITÄT
ILMENAU

Supervisor:
TOBIAS SCHWANDT, M. SC.

Advisor:
PROF. DR. WOLFGANG BROLL

Table of Contents

List of Tables

List of Figures

1. Introduction

The need for more interactive medium is now a necessity for which Mixed Reality (MR) can be a promising solution. Mixed reality is the merging of real and virtual worlds to produce new environments and visualizations, where physical and digital objects co-exist and interact in real time. Mixed reality does not exclusively take place in either the physical world or virtual world but is a hybrid of reality and virtual reality. Augmented reality, a related term, takes place in the physical world, with information or objects added virtually. [1]

The basic idea behind this project is to develop an immersive platform which provides real-time interaction between users in an augmented and virtual environment and would also be able to collaborate between users using different platforms to participate in the Mixed Reality environment. To achieve our goals, we have defined our project goals as following:

- Multi user (multiplayer) environment
    - Multi user AR environment
    - Multi user VR environment
- Collaboration of both environments
- Virtual avatars
- Interactions with users (text, speech)

1.1. State of the Art

The increasing in technological advancements and the desire of effective conferencing systems is having a tremendous impact on how the people used to work. Specially during the times of Covid-19, economic and other environmental reasons such as transporting people for physical meetings consumes lot of energy, time, and money, which these days is another crucial factor for the adaption of new work methodologies. More people are working remotely, more people want flexible work options, and more people are re-thinking what it means to be physically present. This is where the virtual meeting platforms come in.

Virtual meetings are meetings that happen online rather than physical, with all the participants in the same meeting room which can be either teleconferencing, video conferencing or web conferencing. People typically use virtual meetings for work, to involve remote team members from around the world or being used as learning methods in various fields of life [2].

The problem is, people who regularly use interactive platforms, tend to acquire more than just the need to be physically present at a desktop machine to conference. Limitations of current systems include the lack of spatial cues, the difficulty of interacting with shared 3D data [3]. Especially when it comes to students the potential disadvantages of videoconferencing technology are difficulty in sustaining the interest of the learners, lack of training and guidance for teachers and/or students [4].

A teleconferencing application ACME (Augmented Collaboration in Mixed Environments) was developed which used mixed reality to enhance a collaborative teleconference. Augmentation of virtual objects and avatars based on marker tracking, head tracking interaction and monocular camera for hand tracking and interaction with virtual objects using hold and grab gestures [5].

A 3D live interaction system for mixed reality was developed. The system allowed video conferencing between the users to interact with each other as live holograms by using pose estimation in real time [6].

The need for physical presence enhances the surrounding environments to be more understandable and interactable whereas the current teleconferencing systems are very much biased towards transmitting video.

1.2. Project Scope

The game will be made using the popular Unity3D gaming engine (2020.3.14f1). Unity allows you to write code either in JavaScript or C. We will use C# on Visual Studio 2019 due to its robustness and as we have experience in it. While Unity will handle all the physical aspects of game from bringing everything together to the end product, Photon is used for the networking and real time interaction between users in the environment which includes Voice chat and text chat. On the other hand, to participate in the application, for VR we use Oculus Rift S which has slight advantages of comfort and performance over HTC Vive [7] and for AR, an android phone as the development was done using android Unity platform. For VR we are using the XR Unity Toolkit [8] and for AR we are using AR Core by AR Foundation [9].

1.3. Environments

The project focuses on providing Mixed Reality so the users could be at ease to participate having different options. For this, low-poly environments [10] [11] are created using Unity which will provide user to select the desired environment, roam, interact and communicate with different entities in the different scenes. The project is tested with android and Oculus Rift S only however, it should work with other's because Unity's XR Interaction is used, and it offers cross-platform support for VR Development.

1.2.1. Virtual Environment

Virtual environment refers to an artificial environment in which users can interact around the immersed surroundings. In our project, we created the virtual environments using XR-Rig for users to switch in between. The user will be able to see around objects using head tracking with the help of Oculus Rift S and interact with them. We first create this environment in VR and then convert it to AR so the difference is the sky box.

A skybox [12] is basically a sky dome Figure 1, which is a method of creating backgrounds to make an environment appear bigger. When a skybox is used, the level is enclosed in a cuboid.

Figure 1: Virtual Environments

1.2.2. Augmented Environment

Augmented reality is enhanced version of the real world achieved by computer generated elements, overlayed in real environment Figure 2. ARCore is used for the augmented environment. To roam around the augmented environment, we use AR-camera. The user can use an android phone to participate and interact with other users in the application.

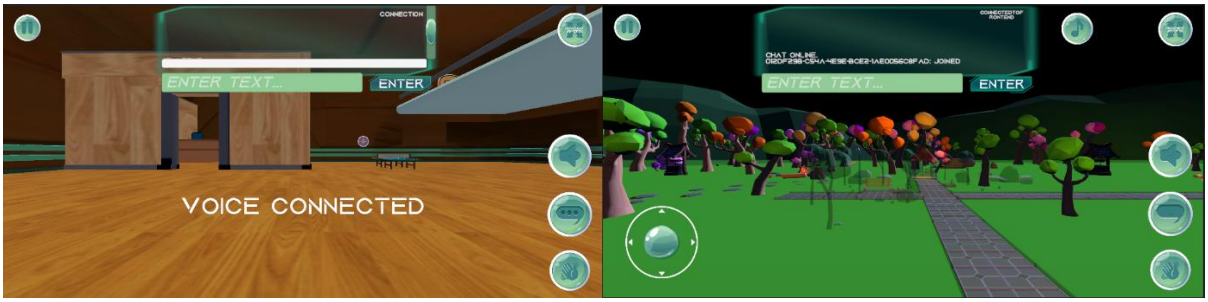


Figure 2: Augmented Reality Environments

1.2.3. How does it work

We made two scenes in the game and then we collaborate this application through Photon load-level [13] which basically synchronises the application between different users. So, when a user opens a scene, the user is just initiating himself in the scene where every user is available.

1.4. Avatars

Avatars define a particular animated model which represents humanoid layout or any character with the capability of physical motions. We have different options to select between avatars to provide each user with a unique physical identity. The avatars can be either anonymous or specific usernames which will pop on top of their head so that users present in the scene can identify and interact with individual users of their choice. The avatars we use are low-poly so they can work perfectly in the application. We use low poly character so there are less latency in online as our project support AR and VR so we wanted it to be both platform friendly. this avatar have no legs so it kind of a robot. the idea behind having no legs is to use it easily in this project without implementing script just to walk which make it easier to use. in VR the hand and head movements are done by using Headset and controller movement. and in AR doing this hard so only made one animation in game wave in VR but body and head movement is done by mobile movement and joystick as well.

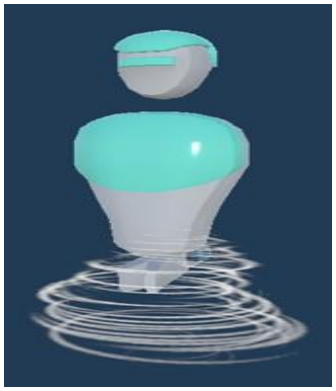


Figure 3: Avatar

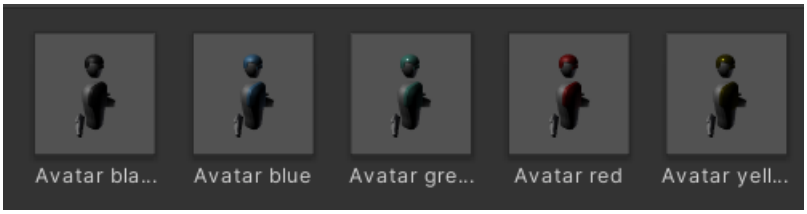


Figure 4: Avatar options

1.5. Photon Network

The Photon Unity Networking [14] also called (PUN) in unity asset store is used as the multiplayer platform, a cloud-based networking implemented in Unity [15]. PUN has several advantages among different networking implements. PUN is a fully integrated with the Unity game engine. It provides flexible API for developer to use their network service and is a stable and resourceful distributed cloud-based server. It does not require hosting from its user. Any player can freely login and leave the environment without interrupting the whole environment. The players can have conversations which are present in the same environment. The players can hear and talk to each other with voice or chat. We found Photon a lot easier compared to other platforms [16] and is rated the highest vote as for game engine.

Other similar platforms include:

- **Unity MLAPI**

  Unity Mid-level API is a framework for building network games. Network having low level access to core therefore providing high level abstractions. Its aim is to remove repetitive tasks to reduce excess code no matter how much modular features we use.
- **Game Sparks**

  A cloud-based backend feature tool for building game server-side features and managing them after launch. It is provided by Amazon but practically dead because SDKs won't run on newer versions of Unity and also has a limited platform and service support [17] .
- **Microsoft PlayFab**

  PlayFab [18] is Photon's competitor [19] with a platform-as-a-service provided by Microsoft built specifically for live games, with player accounts, virtual goods and real money e-commerce, analytics, leader boards, in-game messaging and more. PlayFab SDKs offer easy integration with Unity, Unreal, Xamarin and Cocos. Photon has more customers in game development, game design and mobile games industries.

- **App Engine**

  Google App Engine presents a highly scalable and reliable gaming implementation on Google Cloud Platform that uses Google App Engine and Google Compute Engine for real-time player interactions [20]. The solution powers core game elements such as game matchmaking and player customization by using Google App Engine, while also using Compute Engine to run dedicated game servers and common game engines.

- **Firebase**

  Firebase is another platform provided by Google to create mobile, web applications and offers matchmaking services similar to Photon [21]. The Firebase Realtime Database lets you build applications by allowing secure access to the database directly from client-side code but has the drawback that the solution works slowly if the database model consists of many nested objects.

2. Requirement Specification

The Software Requirement Specification helps in laying out the functional and non- functional requirements of the developing application. The requirements specification defines the foundation of the application, what it will do and what it will not do. It also defines how it will interact with the users with the help of the use case diagrams.

Requirement specifications are divided into two parts Functional requirements and non-Functional requirements. Functional requirements are explicitly defined by the users what he wants, how the system should work. Non-Functional requirements are implicit. They are not defined by the users, but designers have to keep them in mind.

2.1. Functional Requirements

Our application should perform all the functionalities that are listed below.

By launching the application, user will see a menu.

- Menu provides Username options and Anonymous options
- By entering username and connect user is directed to lobby
- By selecting avatar and scene user is connected to the application
- By tapping on pause menu user can pause or go back to lobby
- By tapping on mute application sound will turn off

2.2. Non-Functional Requirements

- Availability
    - Once the application is installed it can be used anywhere and anytime by the user
    - The application should be able to host maximum 20 users at a time
    - The application supports HMD (Oculus Rift S) and android devices
- Efficiency
    - User can communicate in real time
    - User can chat in real time
- Usability

  When the user wants to use the application, the response is rapid
- Reliability
    - Application must be reliable by showing exact usernames and avatars
    - Voice and chat quick response without any delays
- Security

  Application should be secure by allowing the user data to be discreet which includes
    - Voice data
    - Chat data
    - User location

2.3. Use Case

The use case diagram basically illustrates most of the functional requirements of the application and an overview of how the users will interact with the application once it is built and put into use. Every use case represents a different task or event that our application will do. The main use case diagram of our application is shown in Figure 5. The actors in this case are the persons who will be using the system which are our general users.

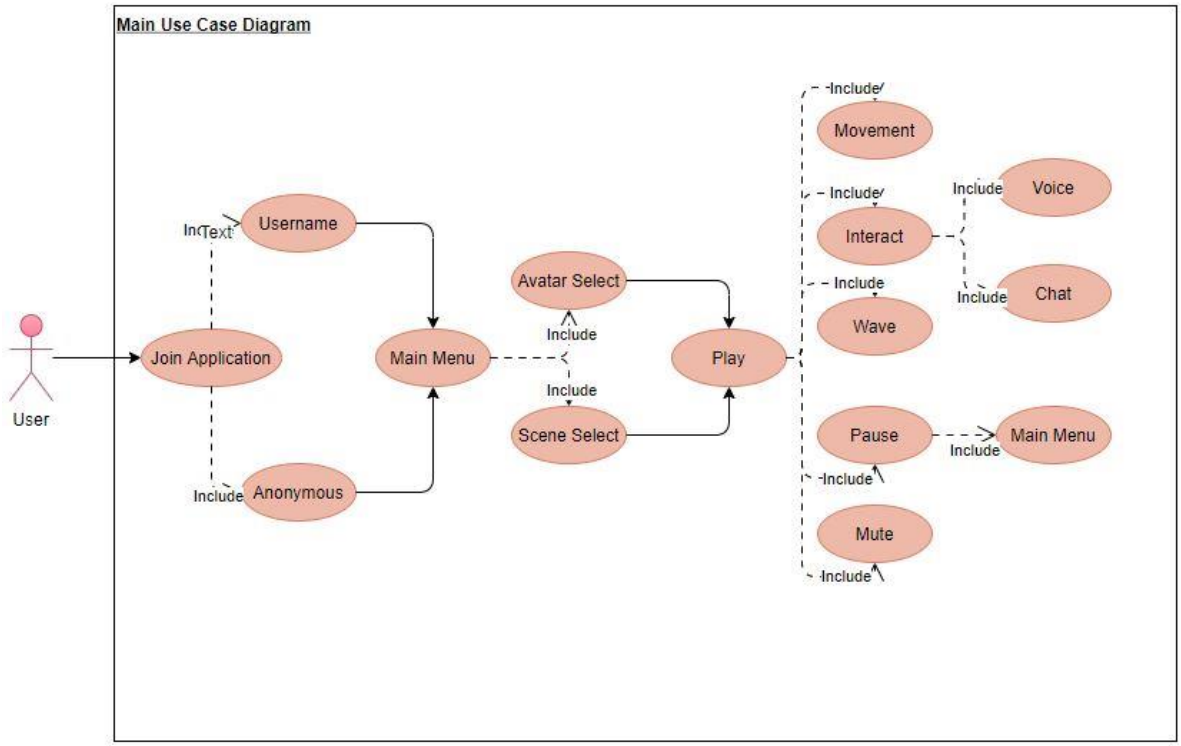


Figure 5: Main Use Case Diagram

The individual use cases are shown below, starting from Figure 6 to Figure 10 with their respective specifications table.

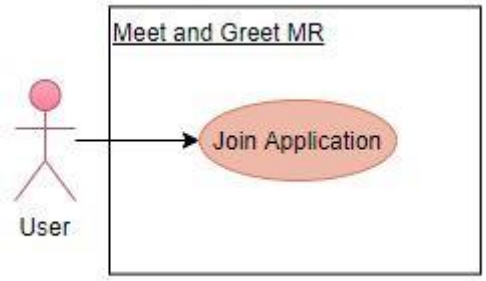


Figure 6: Use Case 1: Join Application

Use Case ID	UC-1
Title	Join application
Actors	Users

Pre-Conditions	Internet, HMD, Android phone
Post-Conditions	Username menu will be opened
Assumptions	User has tapped on connect anonymous or enter name button
Description	• User can select to join as anonymous • User can enter a specific name
Trigger	Main menu (avatar and scene selection menu) will be opened
Main Success Scenarios	User has tapped join and enters the main menu

Table 1: Use Case: 1: Join Application

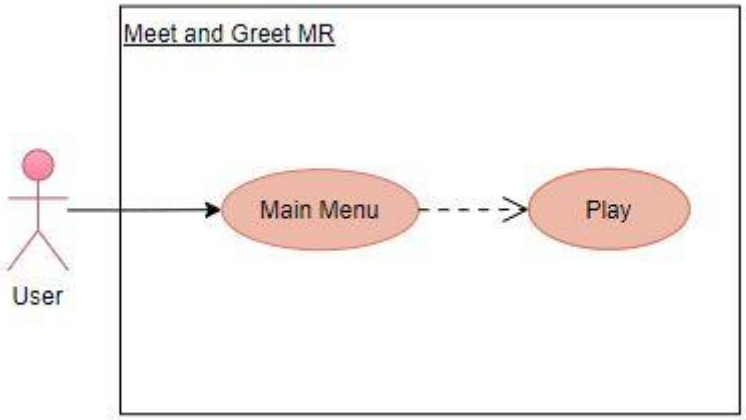


Figure 7: Use Case 2: Main Menu

Use Case ID	UC-2
Title	Main Menu
Actors	Users
Pre-Conditions	User has entered username or connected as anonymous
Post-Conditions	Main menu will be opened
Assumptions	User has tapped on join
Description	• User can select avatar • User can select a scene to enter

Trigger	Specific scene will be opened
Main Success Scenarios	User has selected the avatar and tapped enter on the specific scene

Table 2: Use Case 2: Main Menu

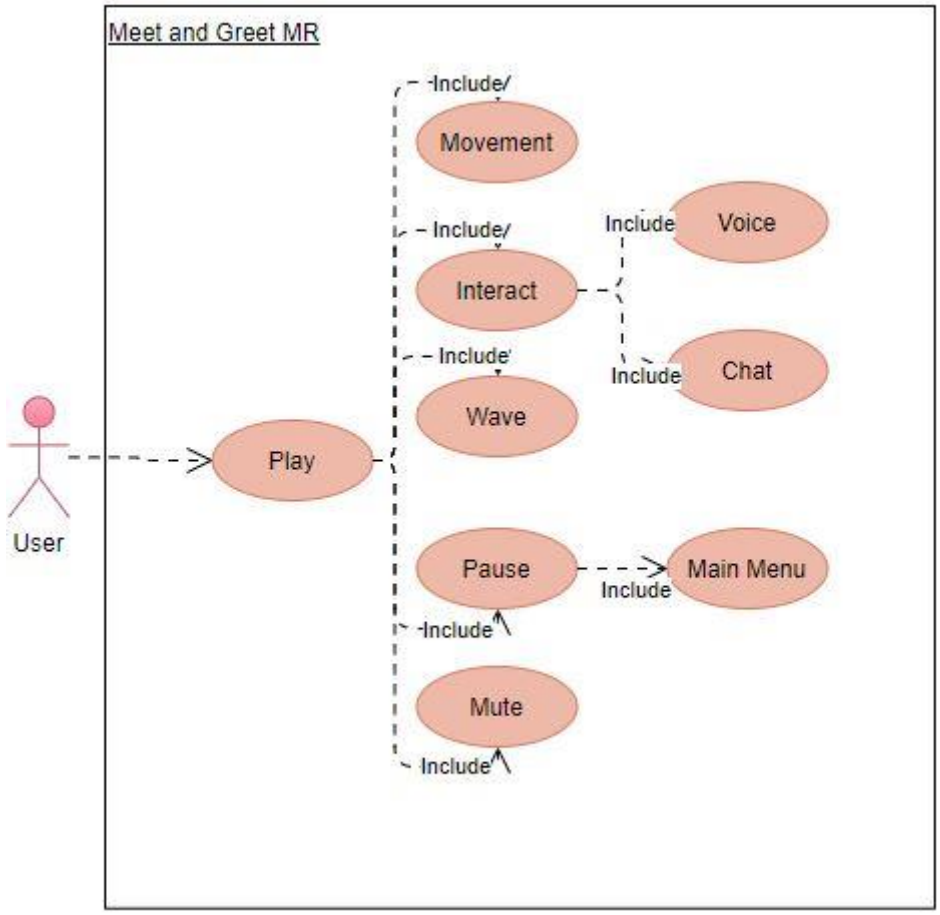


Figure 8: Use Case 3: Play

Use Case ID	UC-3
Title	Play
Actors	Users
Pre-Conditions	User has selected the avatar and an environment
Post-Conditions	Specific environment will be opened
Assumptions	User has tapped on enter button

Description	When the user enters a scene they can do the following in the application on defined devices. Oculus Rift S: • User can teleport • User can communicate Android: • User can communicate • User can wave • User can move with joystick
Trigger	Playing, interacting environment will be opened
Main Success Scenarios	User has tapped enter on an environment

Table 3: Use Case 3: Play

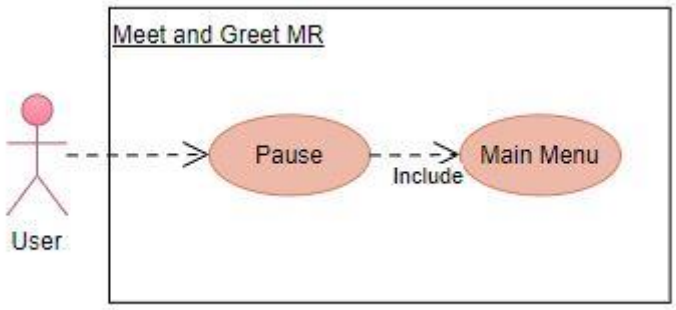


Figure 9: Use Case 4: Pause, Return

Use Case ID	UC-4
Title	Pause
Actors	Users
Pre-Conditions	User has entered the playing environment
Post-Conditions	Application will be paused and home button appears
Assumptions	User has tapped on pause button
Description	• User can pause the application

	• User can return to main menu
Trigger	New menu will be opened from which user can return to main menu
Main Success Scenarios	User has tapped pause button

Table 4: Use Case 4: Pause

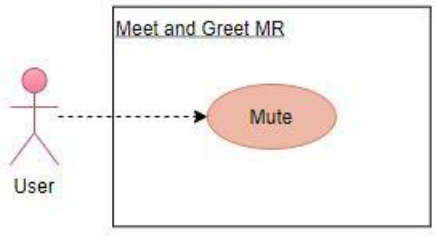


Figure 10: Use Case: Mute

Use Case ID	UC-5
Title	Mute
Actors	Users
Pre-Conditions	User has entered the playing environment
Post-Conditions	The sound will be muted when participating in the environment
Assumptions	User has tapped on mute button
Description	User can mute the sounds of other users in the application
Trigger	The sounds in the environment will be disabled
Main Success Scenarios	User has tapped mute button

Table 5: Use Case 5: Mute

3. Experimental Setup

Our approach to this project is to first create different environments for the users to operate in and then make them compatible to work as augmented and virtual reality for which Photon engine is used. Figure 11 shows the workflow [22] our project and the main objectives:

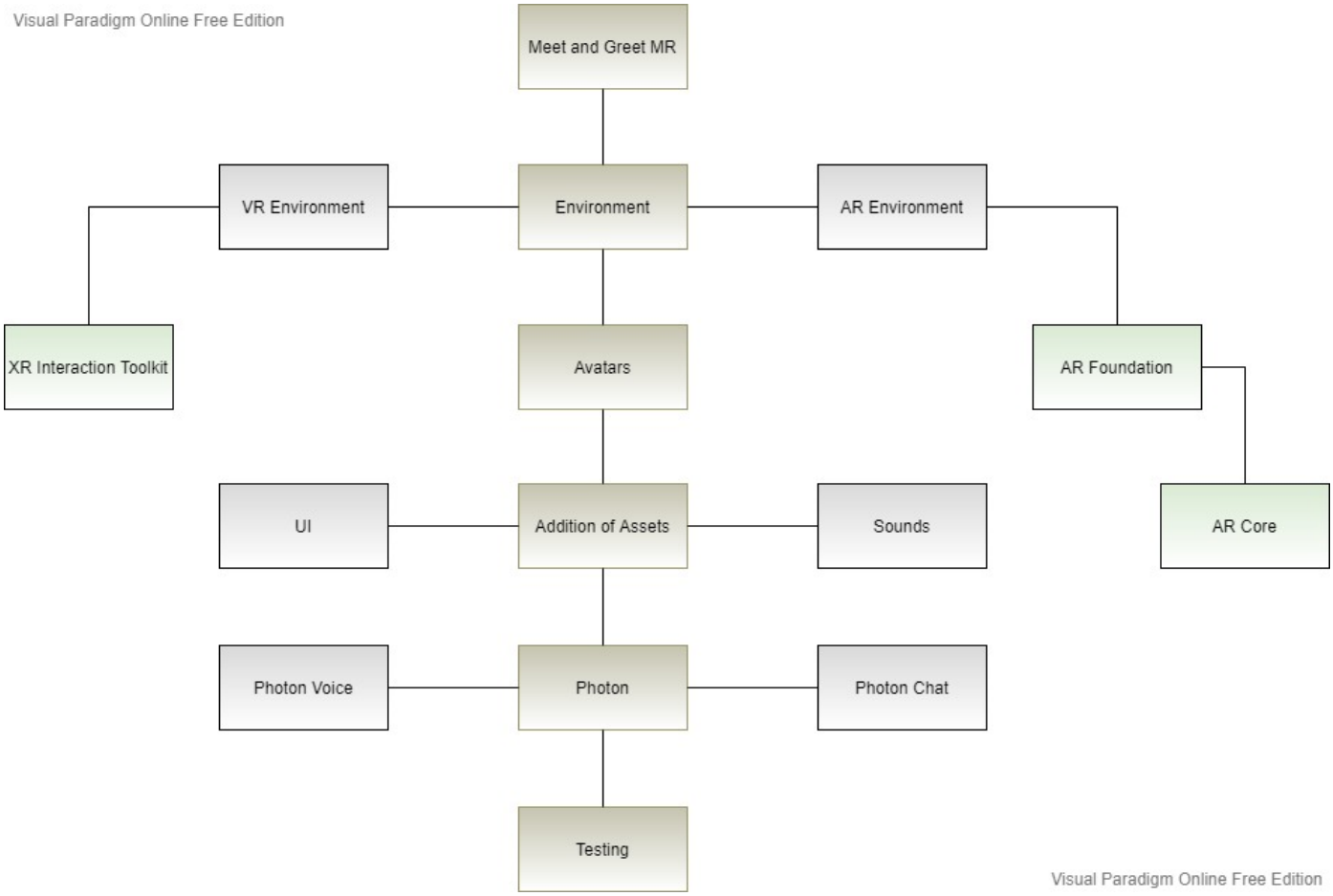


Figure 11: Workflow

3.1. Avatar Implementation

To represent the user, we have an option to select between options of avatar models. We are using low-poly avatars to increase the performance and synchronization in the application. Previously we tried using high-poly avatars which were resulting in deformation of model while interacting.

The user has the option to select his avatar from the provided avatar options after he has connected to the application. The selection window and a mirror appears Figure 12 so that the user can have the desired avatar. The selection can be done after registering in the application as a user or anonymous.

Figure 12: Avatar Selection

Avatars are implemented to simulate body movement based on head movement. For this we need to synchronize the VR transforms with avatars through script and then link the avatar to the script for the transformations to work properly.

3.2. Movements

To explore around the environment, we must keep in mind the different platforms being used in our application. For this reason, the movements are entirely different (in terms of platform used). If the user is connected through an HMD the movement scenario is different to when a user is connected in an augmented environment using a camera (android phone).

The area to explore the Virtual room is always limited and so for this we have set teleportation and moving with joystick which will only aid the user to roam freely without much physical interaction. The movements vary from Virtual reality platform to Augmented reality platform. In VR we use the Oculus controllers while on the AR platform we AR-camera and joysticks to roam around the environment. We set up and divided these movements in two parts as defined below.

3.2.1. VR Movements

3.2.1.1. Continuous

To move around we need to move XR rig in the scene. Movement in VR is also called locomotion. XR interaction toolkit provides sample scripts to implement locomotion easily.

$$(Add\ component \rightarrow XR\ rig\ \rightarrow Locomotion\ system\ script)$$

We need to get inputs from the controllers to be able to move so we add default input actions from the XR interaction toolkit and use continuous move provider-action based.

3.2.1.2. Character Controller

$$(Add\ component \rightarrow XR\ rig \rightarrow Character\ Controller)$$

Character controller component is a default component that allows us to constrain the movement by collisions without having to deal with rigid body. It helps the VR player to collide with walls and obstacles.

3.2.1.3. Teleport

Teleport in VR is to prevent motion sickness. We can use it by adding:

$$(Add\ component \rightarrow XR\ rig \rightarrow Teleportation\ provider)$$

Through this we can move our XR rig to desired location on where we choose. These controls are defined in game controllers Figure 13 and lobby controllers Figure 14:

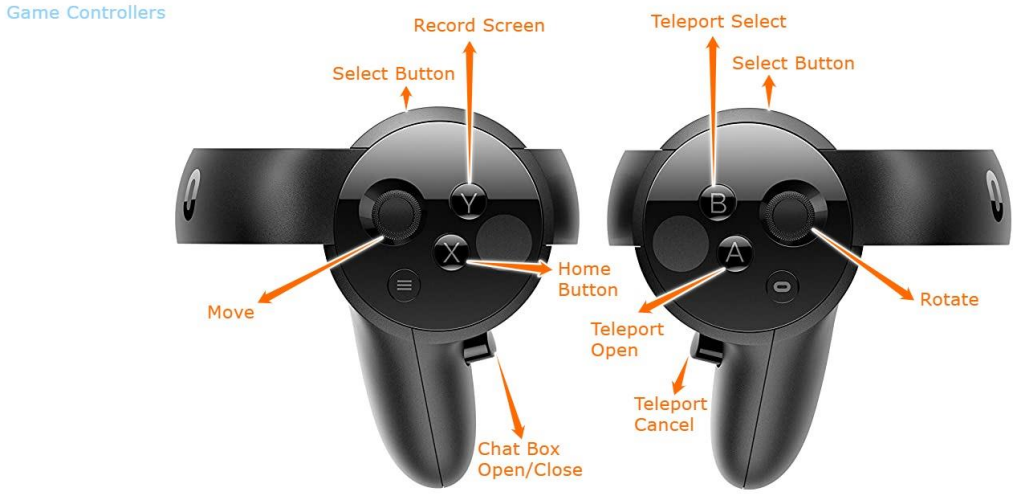


Figure 13: Game Controller

Lobby Controllers

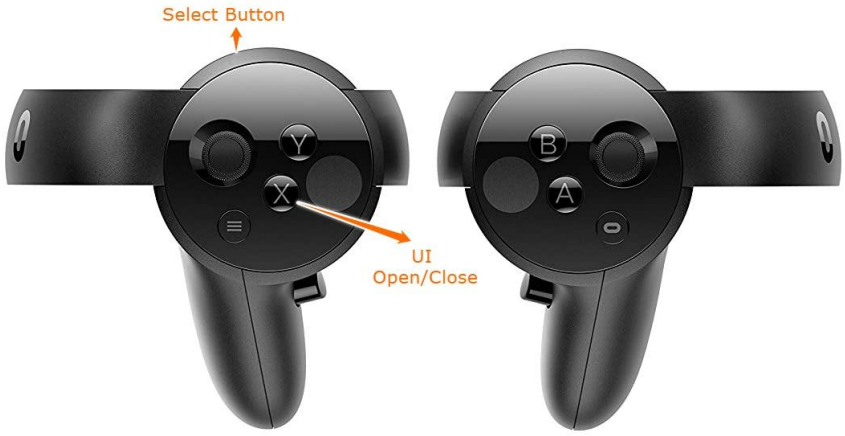


Figure 14: Lobby Controller

3.2.2. AR movements

The movements in case of augmented environment are the same as in VR but differ through actions. As in AR we use the AR-camera to explore around but in VR we can move through Oculus controllers or teleport. We created buttons for augmented environment, so the user has the freedom of interacting and roaming with physical actions. These are shown in Figure 15 and defined below:

Figure 15: AR Controllers

3.2.2.1. Joystick

When participating through the AR platform (using android), there could be a need to participate without physical interactions. Of course, the AR-camera is there but still if there is user disability or motion sickness issue the user can use the buttons instead. The joystick helps the user to move around the environment just like walking around.

3.2.2.2. Play/Pause

The pause button is considered a necessity for applications. However, in the case of a virtual meeting it might seem of no use but still one cannot ignore the possibility of a break or some important stuff to do while in the application.

The button also provides another menu (Home) which migrates the user back to main menu so that if user wants to make some changes to the avatar or change the environment they don't have to restart the application.

3.2.2.3. Sound

The sound function is used the mute the voices from other users in the environment. While talking to other users or participating in the voice chat the user can use this button to mute the sounds.

3.2.2.4. Show/Hide Joystick

The button again is provided to hide the joystick button in case the user wants to navigate physically or they don't feel the need to see too much interface options on their screens. It is provided as an ease for the users.

3.2.2.5. Rotation

For the rotation around the environment the rotation option is provided in case the user is not able to participate physically. They can just sit in one place and use the option to explore around the environment.

3.2.2.6. Microphone

The microphone is an important point to remember so that users can mute their voices in a voice chat. So if a user is not feeling to participate and still wants to hear others, they can mute their mic and continue.

3.2.2.7. Text chat

The button provides the option to chat in text. A chat window appears in top of the screen where users can send and view messages from other users.

3.2.2.8. Wave

Apart from the application functionality we also have to keep the user interest in the application so they might not find it annoying or boring. For this a wave gesture is added as a hello or greetings action for a little joy for the user. When pressed the avatar waves his hand and the users around can see.

3.2.3. Screen Recorder

Screen recorder can be considered as a non-functional requirement but sometimes it is more likely a functional requirement. We have an option for recording the screen as it can be beneficial sometimes when one must keep the track of collaborative meetings. The screen recorder can be started with “Y” on left Oculus controller.

3.3. Photon Engine

To join the environments the avatars added, which are a physical representation of each user. The users will be able to choose between different looking characters of their choice and use Photon for the application to work as multiplayer.

There are two main concepts in PUN: rooms and lobbies. Connecting to a lobby conceptually corresponds to connecting to a session server, while connecting to a room corresponds to connecting to a game server [23]. This means that a player must connect to a lobby before they can connect to a room. After successful authentication, the player joins a lobby from which they can join a room. Photon Servers do not simulate the game. They only provide a means of sending data from one client to the other that fully simulate the game and use PUN to synchronize the game state between themselves.

Photon makes the scenes compatible to work as Augmented and Virtual Environment. Each user will be able to interact and see the avatar of other user present is the specific environment through Photon server which is illustrated as in Figure 16:

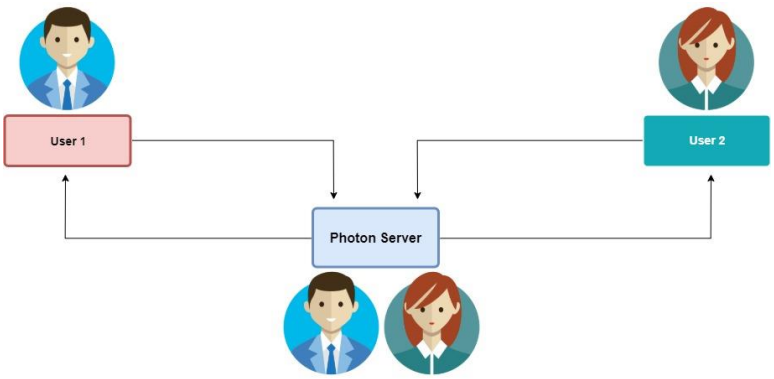
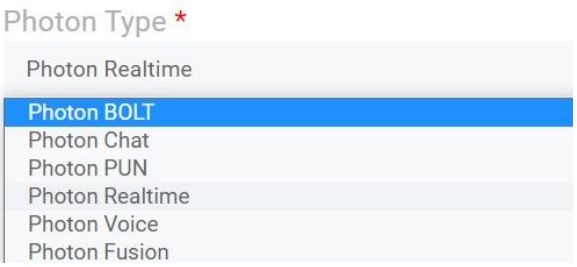


Figure 16: Networking

The users can have different avatar, username or present himself as anonymous. The connection setup with Photon is relatively easy.

3.3.1. Connection Setup

The connection setup for Photon is relatively easy. We need to register for Photon and create an app of our use with the specific Photon type. For our project we will be using Photon Voice and Photon Chat.



Importing the *Photon PUN 2* package from the asset store into the project directs us to the PUN Wizard like in Figure 17. Alternatively, it's in the menu: "Window", "Photon Unity Networking" where we can mention the AppIds created on the Photon Cloud. PUN uses a Realtime App Id for each connection (Voice and Chat).

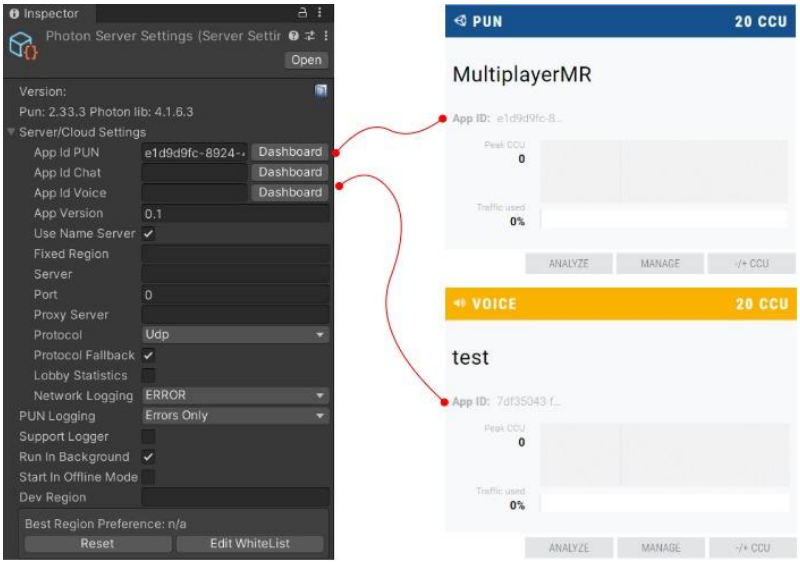


Figure 17: Photon Server Settings

4. Design

In this chapter we will discuss the architectural details of our proposed application. The basic components and the main modules of the system, Graphical User Interface (GUI) and the external interfaces required for our system are discussed. Our world represents a realistic simulation of face to face meetings. It contains the most necessary elements of an interactive medium. The design for the developed application comprises of different modules. System Architecture is given in Figure 18.

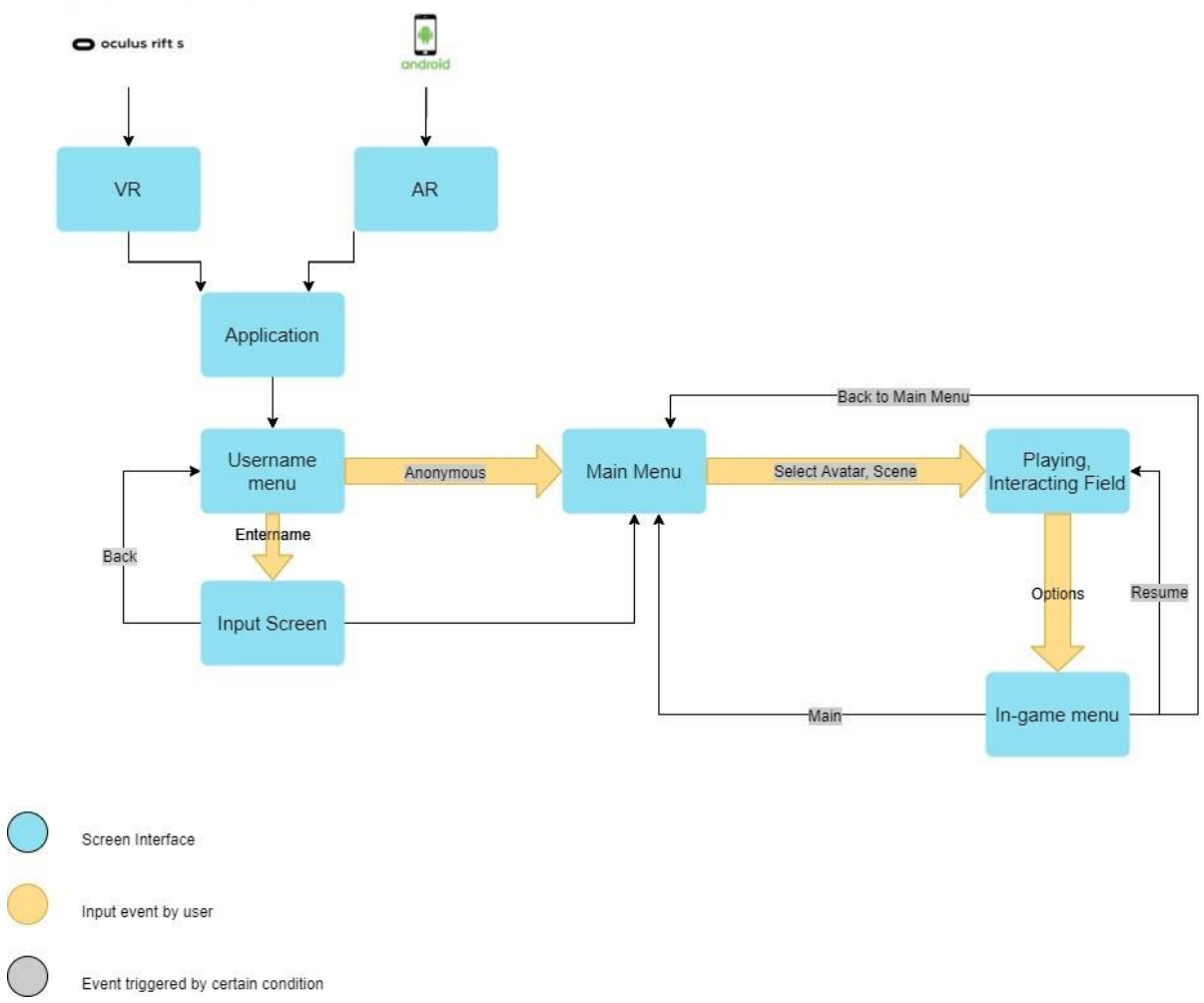


Figure 18: Application Architecture

4.1. User Interface

The user interface has logically two sections: the main menu and the in-game areas. The user interface is designed to be simple, with a focus on making its use a quick and easy process for users with different hardware. When participating through a VR device there is a slight difference in environment (skybox) as compared to participating through AR. For this we have pointed out the difference and interface of each platform when using the application.

4.1.1. AR Interface

The AR interface provides buttons for movements and specific actions and overlay of the environment in the real environment.

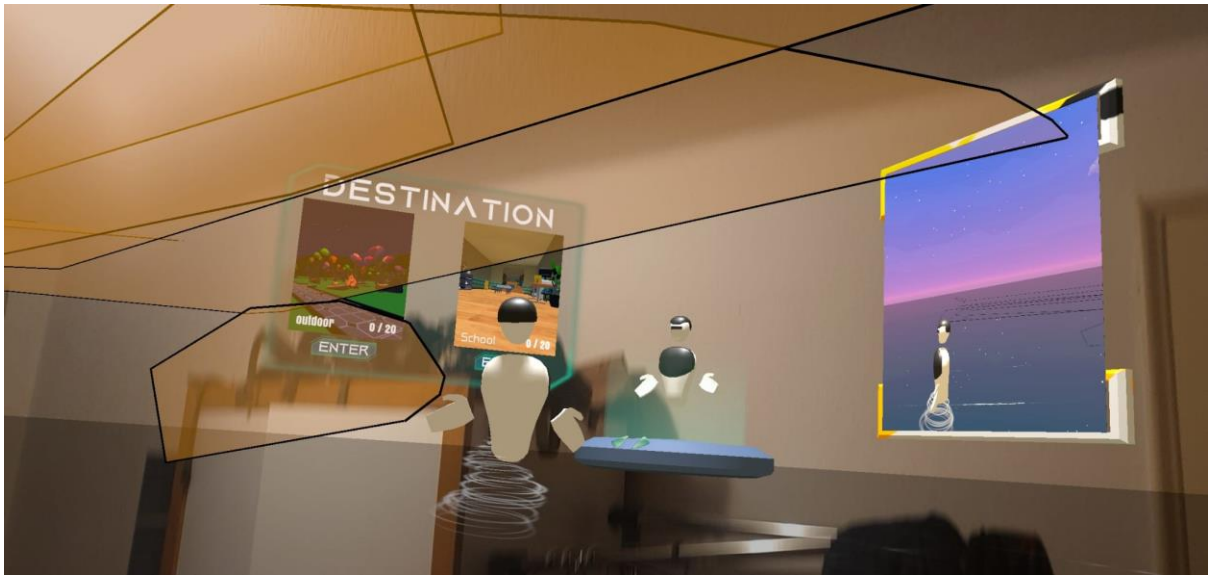


Figure 19: Main Menu

There are different buttons for different actions like pause, mute and joysticks for movements. The buttons are shown in the playing environment below.

Figure 20: Play Interface

4.1.2. VR interface

The virtual interface is a slightly different Figure 21 from the augmented one. In the environment we have a skybox, and the difference is also obvious as in terms of the definition of AR and VR.

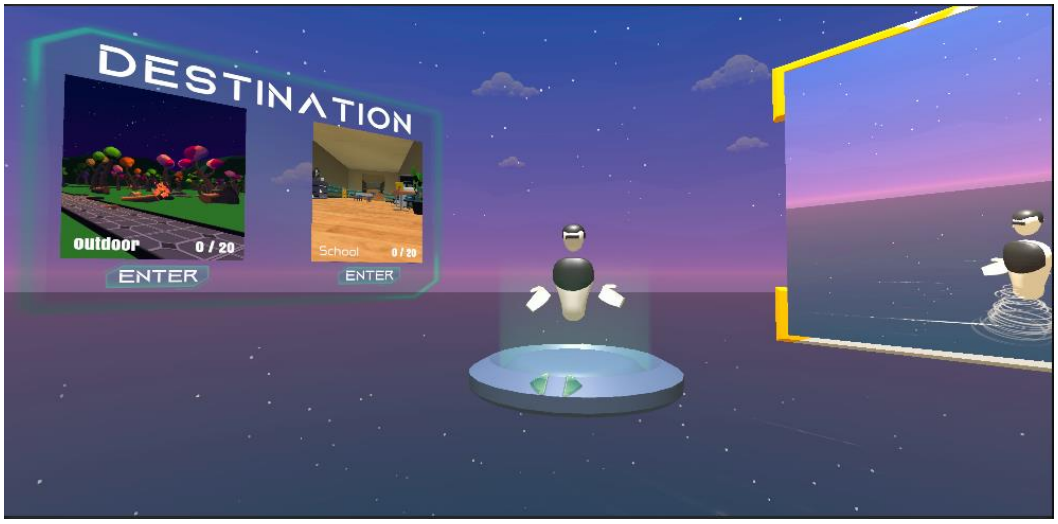


Figure 21: VR Environment

4.2. Audio Design

In an audio design phase, we design the audio of our application to give our users an exciting feel while they are waiting in the lobby or whether they are participating in the environment. For this we selected two audio files from the Unity Asset store which are divided background music and environment music integrated with the button in the UI.

For background music we use (The Intergalactic Beets Project) from the asset store [24] and for the in-play music we use the Atomic Rush [25] music also provided by the asset store.

5. Summary

The main aim of the project was to create a mixed reality environment featuring real time interaction between the participating users. The users are presented with different avatars for the aim of unique identity for each. The application focuses virtual reality and augmented reality platforms. For the application to run virtually we used Oculus Rift S virtual reality headset and an android phone for AR-camera tracking for the augmented reality platform. Unity 3D is used for the whole development process which is includes Virtual and Augmented development, character setup, interactions, and communication.

For communication between the participants, we use the Photon server. After a thorough search of different platforms, we found photon engine to be an easy integration with Unity and reliable source for our application. Also, photon engine has the high reputability in the gaming category as we found in our research. In our interest, photon engine also provides a free version which can support up to 20 concurrent users. The communication is done via Voice and Chat in the application.

References

[1] 21 September 2021. [Online]. Available: https://en.wikipedia.org/wiki/Mixed_reality .

[2] R. C. Team, 21 February 2021. [Online]. Available: https://www.ringcentral.com/us/en/blog/virtual-meetings/.

[3] A. J. Sellen, "Speech Patterns in Video-Mediated Conversations," in *Conference on Human Factors in Computing Systems*, Toronto, 1992.

[4] M. Martin, "Seeing is believing: The role of videoconferencing in distance learning," *British Journal of Educational Technology*, pp. 397-405, 2005.

[5] T. Kantonen, C. Woodward and N. Katz, "Mixed reality in virtual world teleconferencing," in *2010 IEEE Virtual Reality Conference (VR)*, Boston, 2010.

[6] S. J. D. Prince, A. D. Cheok, F. Farbiz and T. Williamson, "3-D live: Real time interaction for mixed reality," in *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, 2002.

[7] R. Tabari, "Laptopmag," 21 March 2020. [Online]. Available: https://www.laptopmag.com/features/oculus-rift-s-vs-htc-vive-cosmos.

[8] Unity, "Unity Documentation," Unity, [Online]. Available: https://docs.unity3d.com/Manual/XR.html.

[9] Unity, "Unity Documentation," Unity, [Online]. Available: https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1/manual/index.html.

[10] Polygon-Park, "Unity Asset Store," Polygon-Park.de, 8 October 2019. [Online]. Available: https://assetstore.unity.com/packages/3d/environments/low-poly-park-154815.

[11] 3. Fantacy, "Unity Asset Store," 3DFancy.com, 9 August 2016. [Online]. Available: https://assetstore.unity.com/packages/3d/props/exterior/lowpoly-wells-68437.

[1 Wikipedia, “Wikipedia,” [Online]. Available:
2] https://en.wikipedia.org/wiki/Skybox_(video_games).

[1 “Photon,” Photon Engine, [Online]. Available: https://doc-
3] api.photonengine.com/en/pun/v2/class_photon_1_1_pun_1_1_photon_network.html#aedba3c40295e3684fedc64acf678a5cd.

[1 Photon, “Photon Engine,” Photon, [Online]. Available: https://doc.photonengine.com/en-
4] us/pun/current/getting-started/pun-intro.

[1 T. Bertorelli, B. Thomas and J. Fujikawa, “Wilderness Survival: Multiplayer Survival
5] Game Using Unity Game Engine C#,” California Polytechnic State University, San Luis Obispo, 2015.

[1 J. Tucker, “Medium,” 31 May 2018. [Online]. Available: https://john-
6] tucker.medium.com/unity-exploring-multiplayer-solutions-20000ccab3b6.

[1 Capterra, “Capterra,” Capterra Inc., [Online]. Available:
7] https://www.capterra.com/p/158600/GameSparks/.

[1 Microsoft, “Azure Playfab,” Microsoft, [Online]. Available:
8] https://docs.microsoft.com/en-us/gaming/playfab/what-is-playfab.

[1 SLINTEL, “SLINTEL,” [Online]. Available: https://www.slintel.com/tech/game-
9] development/playfab-vs-photon.

[2 Google, “Google Cloud,” [Online]. Available: https://cloud.google.com/appengine/docs.
0]

[2 SaaSHub, “SaaSHub,” [Online]. Available: https://www.saashub.com/compare-firebase-
1] vs-photon-engine.

[2 “Visual Paradigm,” [Online]. Available: https://online.visual-
2] paradigm.com/w/xdhpobkt/drive/#diagramlist:proj=0&new.

[2 I. Mekterović and D. Polančec, “Developing MOBA games using the Unity game
3] engine,” in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, 2017.

[24] I. Beets, “Unity Asset Store,” The Intergalactic Beets Project, 7 September 2021. [Online]. Available: https://assetstore.unity.com/packages/audio/music/electronic/the-intergalactic-beets-project-full-length-song-collection-i-199984.

[25] N. S. Lab, “Unity Asset Store,” 5 January 2015. [Online]. Available: https://assetstore.unity.com/packages/audio/sound-fx/ui-sound-effects-collection-pack-2-buttons-27803.

Declaration of Independence

Declaration,

This project "Meet & Greet in MR" is a collective effort of group members. Each member provided assistance and contributed in every aspect of the work.

Syed Sameer Kazmi, Matr. No: 62260, was responsible for the design strategy, Work on VR & AR.

Taimoor Ahmed, Matr. No: 60825, was responsible for Report writing and assistance in creating environment and VR.

Umar Irshad, Matr. No: 61389, was responsible for assistance in report writing, AR and testing.

This work was done wholly or mainly while in partial fulfilment of the research degree at Technical University Ilmenau. We certify that the work has not been submitted in the same or any similar form for assessment to any other examining body and all references, direct and indirect, are indicated as such and have been cited accordingly.

Ilmenau,

26.09.2021

Syed Sameer Kazmi	:	62260	_____
Taimoor Ahmed	:	60825	_____
Umar Irshad	:	61389	_____