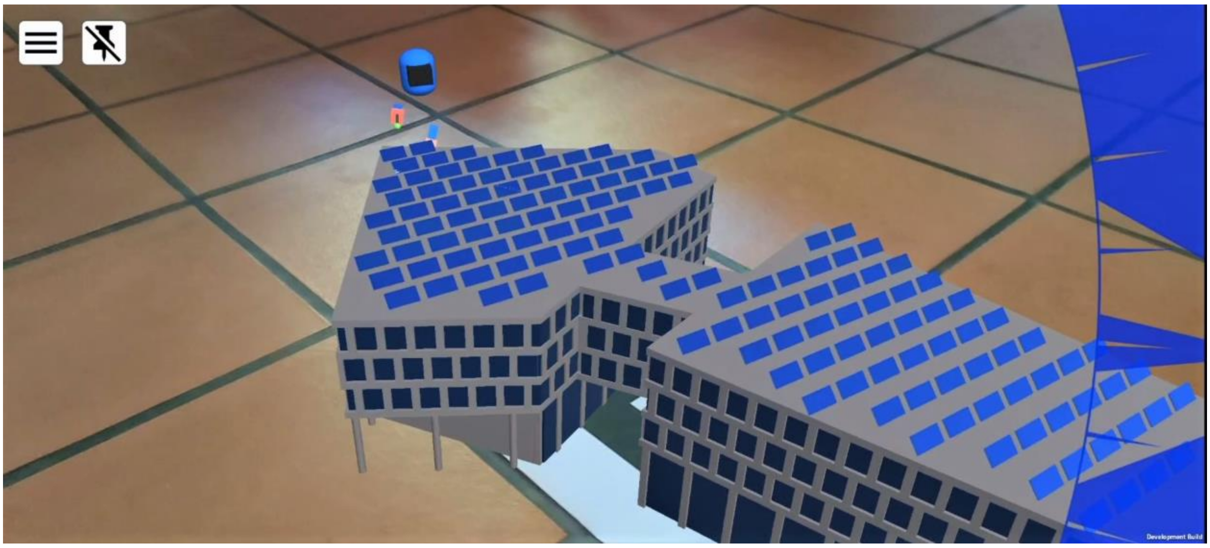
VR-labs on smart devices for education

A Mobile AR Extension to the Virtual Physics Laboratory to enhance the VR Learning Experience



IP5 Project of

Thierry Odermatt
6th Semester
Computer Sciences - iCompetence
thierry.odermatt@students.fhnw.ch

Andreas Leu
6th Semester
Computer Sciences
andreas.leu@students.fhnw.ch

June 17, 2022,

University of Applied Sciences and Arts Northwestern Switzerland
School of Engineering
Brugg/Windisch

Advisors
Prof. Dr Arzu Cöltekin
and Cloe Hüsser

Customer
Prof. Dr Renato Minamisawa
Physics Laboratory FHNW

University of Applied Sciences and Arts Northwestern Switzerland
School of Engineering

Abstract

In this IP5 project, a mobile augmented reality (AR) extension for the existing and in previous projects developed “Virtual Physics Laboratory” (VRLab) was created. For each conducted experiment, only a single user needs a virtual reality (VR) headset, with many additional users being able to join on mobile AR via Android devices. The AR extension allows users to join, observe and interact with the VR user using their own mobile phones, so that the use of expensive VR headsets can be reduced drastically.

The main focus of the development process of this project can be structured into three major parts. First, a mock VR application and an AR application were created. The AR application allows users to join a VR player’s virtual room and observe them. Second, the decision was made to allow dynamic loading and changing game objects, such as the model of buildings. Third, several interaction concepts between AR and VR players were researched, implemented and tested.

The AR application was evaluated in a user study consisting of a task-based workshop and a subsequent questionnaire.

Gender-Neutral Language

In order to respect the linguistic equality of all genders and still follow a consistent writing style, the decision was made to use the generic masculine in this report.

Table of Contents

List of Figures

List of Tables

1. Introduction

In this section, a general overview of the achievements, the user study findings, the initial position and objective of this project and a document reading guide are described.

1.1. Achievements

The goal to create an AR-based extension application for the existing application "VRLab" is achieved. The created mobile application carries the provisional name "AR-Viewer", the VRLabs mock application is named "VR-Player". The purpose of the AR extension is to enable an easy and interactive collaboration between a VR host and many AR viewers. To use the created application, no prior knowledge is needed, meaning that any user can view and interact with the VR player with ease.

Both created applications were created with Unity Engine and written in C#. The VRLabs mock application runs on an Oculus Rift S. Since OpenXR was used, the application should work, with minimal modification, on any VR device supporting OpenXR. The AR application runs on any Android mobile device supporting ARCore.

Using Photon Unity Networking, game objects can easily be synchronised between all users within the same game room and by creating and loading Unity Asset Bundles, game assets can be changed dynamically..

1.2. User Study Findings

The AR application was tested with four persons in a task-based usability workshop with a subsequent questionnaire.

During the usability test, many issues have been identified to be prioritised and counteracted. Interestingly, both testers with, and without prior AR and VR experience could easily learn to use the application.

The general feedback was positive. Users liked the way VR and AR players could be in the same virtual room and that AR devices are visible to each other. Furthermore, it was interesting to observe how the individual testers proceeded and solved the individual tasks. Especially observing the testers' first impression and first steps with the application was interesting.

1.3. Initial Position and Objective

Goal of this project was to create an AR-based extension to the already existing application "VRLabs", which was created in previous projects. The existing application is a VR application that allows users to observe and modify a setup of solar power panel placed on a virtual building. The setup is simulated on a server hosted by the customer. As current hardware serves the Oculus Rift S, due to it being an affordable and already available solution. The Project was originally created on Godot but is currently being ported to Unity Engine. VRLabs uses OpenXR as the XR Interface for VR. The building and solar panel placement is currently done on the Solextron web page. The setup is then saved as a JSON file, which can be transmitted to the VR environment to be viewed. Interactions in VR include only changing the panel type so far. Tilting, moving, adding, or removing panels as well as modifying the building are not yet supported interactions, but will be implemented in the future. The porting process and the implementation of additional VR interactions are not within the scope of this project.

The use-case of the application is within education. The teacher presents the experiments by using the VR headset and displaying them on a bigger display to be viewed. Or he hands out headsets to have the students test the experiments by themselves. The problem is that VR headsets are expensive. Therefore a solution has to be developed to get an inexpensive way to allow students to view and participate in the experiments without needing a headset. Based on this, the primary VR headset-based application has to be extended by mobile device viewers. And possible interaction concepts with the primary user have to be developed.

The AR-based extension “AR-Viewer” adds the ability for mobile users to join a VR host and view and observe the presented experiment. In addition to viewing, the AR user should be able to interact with the VR host. Therefore, different interaction concepts have been researched and found, and a selection of interaction concepts were implemented.

1.4. Document Reading Guide

In the next chapter, “Literature Research”, the relevant terms and findings of the literature research covering “Photon Unity Network”, “Dynamic Unity Assets” and “Interaction Concepts” are explained. In the chapter “Development of the AR-Extension”, a more detailed insight into the development of the AR application is given. To get a more detailed view of the user study, there is the chapter “Usability and User Experience Testing”. The results of the user study and a short analysis are given in the chapter “Results User Test”.

The chapter “Discussion” elaborates on the achievements and results of this project, as well as possible future extensions and a few final words. At the end of this report, in chapter “Appendix”, all relevant documents such as the integration guide, the user study preparation, conduction and findings, the project agreement or the original project description can be found.

2. Literature Research

In this section, the theory of the three big parts of this project will be covered in detail.

2.1. Photon Unity Networking

Photon Unity Networking - short PUN - is a tight Unity package for multiplayer games based on the Photon engine. Photon engine is a game engine created for multiplayer game development. PUN was chosen because it has a great Unity integration, good documentation and is widely used and well tested. It is being used by more than 600'000 studios and developers (Exit Games, 2022c) with a player base of over 20 million online users (Bryan Wirtz, 2021).
PUN allows to implement and use flexible matchmaking such as creating several game rooms for the users to connect to with synchronised game objects, such as the building model. It also supports development by providing dedicated servers and pre-made and optimised networking, so that the AR-Viewer base functionality can be implemented fast and reliable (Exit Games, 2022b).

2.1.1. Basics of PUN

To use PUN, the first step is to connect to the photon network. This can be achieved by using PhotonNetwork.ConnectUsingSettings(). As PUN uses callbacks, on every connection or action that is sent over the network, it can be reacted to its success or failure. Therefore, if the network connection is established, the user can be connected to a lobby and the room list of that lobby can be displayed. In theory, different lobbies based on the user's preferences such as location could be created, but in our use case, all users will be in the same country and therefore connect all into the same lobby. Within a lobby, the user can create or join game rooms and load the game scene. To synchronise game objects such as the building model with its solar panels between the VR-host and the AR-viewers, we can make use of PUNs networked game objects, which will contain a PhotonView to observe changes and synchronise them to all users within the same game room. (Exit Games, 2022b)

2.1.2. Ownership

With networked game objects, a common issue occurs - ownership. If a networked game object is instantiated and two users make changes to this object. If, for example, both users want to move a cube, the object has to decide whose changes it will use. This usually is decided by the ownership. In PUN, each networked object has a creator, an owner and a controller. The creator is the user who instantiates the object, the owner is the default controller of the object and the controller is the user who has the current state authority over the object, which usually is the owner unless he is disconnected. If user A instantiates the cube, he is the creator, owner and controller of it. If user B wants to control the cube, user A has to transfer the ownership to user B. In PUN, this can be achieved by setting the transfer type of the object in its PhotonView. This type can be set as type fixed, where the ownership cannot be transferred, type request, where user B requests ownership from user A and user A has to accept or decline the request by a callback method, and type takeover, where any user can directly claim the ownership of the object. (Exit Games, 2022e) In our case, the AR-viewers are not allowed to make changes to the game objects, so in most cases, the transfer type will be fixed to the creator, the VR-host.

2.1.3. Optimisation

PUN is already well optimised for performance and network traffic, but there are still some points to mention which can optimise an application even further. For this project, the network traffic is one of the most critical parts, as our end-users need to be able to use mobile devices with mobile data. Therefore it is important to only synchronise what is absolutely needed. For example, the building model only needs to be displayed for the AR users, they cannot make any changes to it. Whenever possible, synchronisation of an object will be done one sided to reduce network traffic, as most of the game objects may only need to be displayed in AR. As most objects will only be placed and do not have to be able to scale or rotate in any way, the only data that needs to be sent is the location of the object. This kind of optimisation can be done extremely deep using photons OnPhotonSerializeView, which gives the ability to control the serialised data of the game object.

Apart from network traffic optimisation, there are other performance optimisations like object pooling and caching. For example, networked game objects can be loaded from a prefab pool instead of the default loaded resource, which can be slower as the loaded resource is kept in memory. Another option is to cache more detailed data than the default, which in some cases will help to find referenced objects or components faster. (Exit Games, 2022d) In our case, as not a lot of data gets synchronised and as a small number of actions and models is used, most of the optimisations will not provide many benefits and would not be significantly testable.

2.2. Dynamic Unity Assets

To enable different building models for different VR users, these models - in the following sections also named asset - need to be stored and loaded externally. In Unity, this can be achieved by building Asset Bundles. An asset bundle is game content that can be stored separately from the main application, for example on an external server, and loaded at runtime. Besides allowing users to update and add content at runtime, this also helps to optimise network traffic and system resources, as only assets that are actually needed by the user must be loaded. As asset bundles can be stored externally, there are different ways of loading them, for example by downloading them from a web server. (Unity Technologies, 2021)

2.2.1. Advantages

In this application, as described in section “3.5. Dynamic Unity Assets”, asset bundles are used to store different building models and only download and install the models the user needs within his game room. In a larger context, asset bundles bring many more advantages over storing all assets locally. By using asset bundles, update or post-release content can be added to the application without having to release a new application version. This could include all kinds of content, for example, limited-time promotional event content, downloadable content or special seasonal or event-based themed content. It also creates a more modular project environment, in which game content can be more easily updated or reused. Another example could be the company's branding, such as logos, banners or videos. When the company branding changes, only the relevant asset bundle would need to be updated. As a last example, asset bundles could also be used in cross-platform applications. For example in an application with high-resolution models, on a desktop an asset bundle with a higher resolution could be loaded than if the user is on a mobile platform. (Unity Technologies, 2021)

2.2.2. Disadvantages

Besides the many advantages of asset bundles, there are a few points to discuss. Using asset bundles adds an extra layer of indirection to the project which needs better documentation and a clear structure. Apart from that, the only big disadvantage is that asset bundles need to be built within a unity application, therefore automation and implementation by other tools are difficult to achieve. (psuwara, 2018) In our use case, to export a building created in the web application, a separate asset bundle generator - a unity application - has to be built and the created asset bundle has to be uploaded to the web server. This solution seems not perfect for our use case, but there does not seem to be any other way to load external assets in unity.

2.3. Interaction Concepts

As part of this project, several possible interaction concepts between VR and AR users have been researched and implemented. In the following section, these interactions will be discussed by its benefits and drawbacks.

2.3.1. AR-VR Interactions

In-Game Chat

One of the most widely used language-based interaction concepts in multiplayer games is the in-game chat. Whether purely text, voice or video-based or a combination of these, a chat option gives the players an easy and familiar way of communicating with each other. Each of the three types of in-game chat offers different features and experiences.

Text

Text-based chat, or simply chat, allows users to write messages to each other. This provides the ability for quick conversations and requires less bandwidth than audio or video chat. In general, it can be distinguished between three types of in-game chat: In-game overlay chat, sidebar chat and in-game conversation. (Emily R., 2022)

In-game conversations are implemented differently by each developer. A text bubble based example is used in the game “Star War Galaxies”, a massive multiplayer online role-playing game (MMORPG) (Kayaba, n.d.), where the written text is displayed in a bubble above the player's head for a few seconds, visible to all other players in a close range. Besides just text, these chats can also use animations like gestures and poses or given keywords to communicate. (Ducheneaut & Moore, 2004)

A sidebar chat is usually an expandable column of text messages with details like sender and timestamp and which usually also shows a history of sent messages. This kind of chat can be observed in close to all multiplayer games, for example in the game “Counter-Strike: Global Offensive” (Valve Corporation, n.d.-a) or “Warframe” (Digital Extremes Ltd, n.d.).

The in-game overlay chat is usually either a complete or partial game overlay, for example semi-transparent, and allows users to read and write chat and additionally see information like the user's activity feed, online/offline status of friends or game statistics and news. This kind of interaction is mostly used by applications extending the gaming experience, like “Discord” (Discord, n.d.) or “Steam” (Valve Corporation, n.d.-b), but can also be found within games that do not want to have an expandable side menu or that do not want to lay that much focus on the chat.

Apart from the great benefit of communication, which can create a better immersion and game atmosphere, chat features also come with a few drawbacks. If a chat is used, there is also a need for profanity filters, spam filters, muting, blocking and reporting options of other users or another way of moderation, as otherwise it would not be controllable what users write or how they communicate. (Bryan Wirtz, 2021) In our use case, this would not be such a big problem, as our environment is controlled, but considering huge multiplayer games where complete strangers can write to each other, it can get a dangerous environment, especially if minors are involved. (Ies, 2019)

In our use case with a VR host and AR viewers, text-based chat is a challenging approach. The AR viewer can simply write messages using their touch keyboard, but the VR user can hardly type, as virtual keyboards are slow to write on. One possible implementation would therefore be to just let the AR viewer write so that the VR player can only read and answer with predefined keywords. If the AR viewer is visible in VR, a bubble-based approach would be possible too. We have dispensed with implementing a chat feature as a separate real-time server would be needed, this would go beyond the scope of this project.

Voice

Voice-chat, either in-game or by an additional service like “Discord” (Discord, n.d.) or “Team Speak” (TeamSpeak Systems, n.d.), is besides text-based chat one of the most used interaction concepts for multiplayer games. Communication by voice deepens the immersion of the game world, helps to forge and strengthen social bonds and allows clearer and faster communication than text-based chat, for example, to discuss tactics and strategies. In comparison to text, with voice, the player can still use his hands to play instead of typing, which is found to be more enjoyable and additive to in-game effectiveness and can even influence emotions like fun, suspense or fear. (Subspace Team, 2021) To further increase the immersive experience, some games like “Phasmophobia” (Kinetic Games, n.d.) use in-game voice chat combined with a simulated hearing distance, so that players only hear each other clear and loud if they are within a reasonable distance.

Some drawbacks of voice chat are that the immersion and experience is only better if the audio is clear and without disturbing background noises, if players bring some discipline and respect with them, and if the used network is reliable enough to handle the needed network connection for high-quality audio connection. In addition, the same dangers as in text-based chat occur, only that it is more difficult to moderate voice chat and prevent profanities. The mute, block and report options should be used.

In our use case, voice chat would fit perfectly, as it would enable all players in the same room to talk and discuss with each other. As it is looked at from a learning environment, it can be assumed that almost all users will be within the same room, therefore the application would not benefit as much from this interaction concept as it could. For this reason, a further real-time voice server would be needed and the users can easily use an external service like “Discord” (Discord, n.d.), “Zoom” (Zoom, n.d.) or “Microsoft Teams” (Microsoft, n.d.) to communicate with each other. We decided to not implement this feature.

Video

Video-chat allows users to not only have audio feedback, but also visual feedback from the other users. It can be assumed that video chat also supports voice chat and text-based chat, as most current video-chat services like Discord, Zoom or MS Teams allow that. This concept is less likely to be used in multiplayer games but could make sense in a learning environment as a combination of an online meeting and the application which will be discussed.

With face-to-face communication, the experience of the application is more meeting based and most likely feels more serious than just the application, which can be a benefit if the application is used alike. A further benefit is direct feedback. All users can see the facial expression of the other players - or only the teacher, in our case the VR host - which enhances the experience in a discussion or learning environment.

With this approach, the disadvantages may outweigh the advantages, as on VR, displaying a video feed may be difficult to implement, without disturbing the VR immersion, and on AR, there is not a lot of space for a video feed overlay on a mobile device while still being able to see and interact with the underlying application. Besides that, video transfer, especially high-quality and high bit rate video connection, creates a lot of network traffic, together with audio and chat traffic will get even higher.

Based on the disadvantages and the fact that video chat usually combines video, audio and text chat, this feature would go way beyond the scope of this project.

Polls

Another interaction that could enhance the user experience and the learning environment of VRLabs is the ability of the VR host to create polls. The idea is that the teacher can create a poll on a currently discussed topic and display it to the students either by a sidebar chat as described previously or by a pop-up overlay which will be displayed above the AR application.

With the pop-up version, the teacher could somewhat enforce participation by not allowing the AR viewers to close or skip the poll until answered. This way, the teacher-student interaction could strengthen and the learning experience would be better.

A clear drawback of this feature is that the experience of the application will be restricted by interrupting polls. The AR users will be less able to explore the game world on their own but instead, have to follow closely the lead of the VR host. Depending on the exact environment the users are based in, this can either be a benefit or a disadvantage. Another difficulty would be to find a suitable way for the VR host to create these polls.

In this project, polls were not integrated as a separate data server would be needed.

Question Bubble

Similar to the bubble-based in-game text chat, the interaction concept of a question bubble would allow the AR viewer to ask a question by text and display this bubble above his avatar. Depending on the implementation, this bubble would either stay until the teacher or the creator would delete it again, for example if the question got answered. The term bubble in this context describes the way of displaying the information, not the shape, form or design of the information. By bubble-based it is usually meant that the information is floating above the player's avatar and moves as one with the user.

With the bubble approach, the way the information or question of a user is displayed is more personal, as the identification is made by the character and not for example by username like in a text chat. All possibilities that come with the text chat may also apply to this concept, but the visualisation will always stay above the players head which allows further arrangement possibilities for the UI.

A big drawback of this concept is, that the visualisation of the bubble is only directly visible to all other users except the creator, for the creator, the visualisation needs to be implemented separately.

This interaction concept is not implemented as it would not bring great use for our use case. In our context, the users will all be in the same physical room.

Highlight-Feature

The idea of a highlight feature to set certain parts or areas apart is in presentational context widely used. As a later section describes a feature for AR users to highlight a certain part of an object (2.3.2. Implemented Interaction 1 - Ping-Tap), this concept targets the other way around, so that the VR user can highlight objects. The basic idea is that the VR user can grab and use some kind of marker, for example a spray can or a big coloured marker to spray or mark any kind of surface he wants. Ideally, the colour of the marker can be changed. Anything the user sprays is synchronised with all other players in the game room, so that they can view what areas the VR host means. Together with the highlighting object, a way to remove the marks would be needed, for example a deletion spray or an eraser.

This kind of highlighting can support the teacher when explaining in general. For example, to show a flow direction of any kind, the teacher can mark the direction directly onto the building. This way, instructions and explanations can be supported visually and can stay persistent.

The only drawback could be the implementation. Based on how the marks are implemented and synchronised, application performance and network traffic could be heavily influenced. Therefore, the implementation should be well planned and studied beforehand.

Based on the complexity of the implementation, we decided that this feature would fit great, but not within the available time span of this project.

Write Anywhere

The idea of "Write Anywhere" is that the VR user can write anywhere in the game world, not being restricted to objects or planes. Again, all written text will be synchronised to all AR viewers and can only be deleted by an eraser of the VR player.

In comparison to the highlight feature described in the previous chapter, the user can now also write besides the object he would like to discuss or explain. This offers the opportunity for the teacher to, for example, write questions into space close by the area, where the students should develop the solution.

A clear issue with this interaction is that the AR viewers may not be able to position themselves in a way that they can easily read what the VR player wrote.

Even if the ability to note down stuff anywhere in space could be extremely useful, in our use case it seems to not be able to show its full potential and was therefore not implemented.

Virtual Whiteboard

The interaction concept of a virtual or online whiteboard for remote collaboration gained a lot of attention since the covid pandemic and with it the wide mandatory home office. A virtual whiteboard allows the free writing on a synchronised surface. For the VR player, this could be visualised by an actual model of a whiteboard, on which he could interact with all other users. The AR users would need to implement this feature differently, as it is not very usable to write with the finger on an AR visualised board. Therefore one possible way would be to show the whiteboard as a separate tap in the overlay and leave it to the user to maximise the page to use the full screen size.

Virtual whiteboards enable easy collaboration on a persistent, expandable, digital canvas and allow the teacher to share more theory based information.

Depending on the implementation, it can be difficult for the mobile users to write on a virtual whiteboard. Writing with fingers on mobile is quite difficult, and to write text, the device orientation should be vertical instead of horizontal as in our use case.

An implementation of a virtual whiteboard seems to fit great, but would be a future extension due to time limitation.

2.3.2. Implemented Interaction 1 - Ping-Tap

The concept of the ping-tap interaction is simple. Any AR viewer can set a ping, a marker, anywhere on the building model or on the solar panels to mark a specific location or object. This ping is then shared with all other users. The idea is that AR viewers can easily communicate to other party members if they have a question or want to add input to a specific place or object.

The concept of being able to set a marker, to ping, is a widely used interaction. Not only in multiplayer games like "Tom Clancy's Rainbow Six Siege", a tactical multiplayer shooter, (Ubisoft Entertainment, n.d.) or “Apex Legends”, a battle-royale multiplayer shooter, (Electronic Arts Inc, n.d.) but also in real life. The so-called signal rocket or signal flare is a signalling device with a similar design to a firework rocket. It is widely used in seafaring, or in smaller versions for surfers, climbers and the military. Signal rockets have the simple purpose of attracting attention by firing pyrotechnic signals at high altitudes. This can be done for the purpose of calling for help, but also as a signal to launch or attack. (PZ Bremen/Hamburg, n.d.) Regarding the use of markers or pings in games, there are clearly close similarities - Attract other people's attention by giving a signal.

Let us have a closer look into the ping feature used in “Rainbow Six Siege” and “Apex Legends”. In “Rainbow Six Siege”, pings are numbered with the player's identifier to allow easier callouts between the team members. Depending on the object that is pinged, the displayed icon changes. In the original system, only one marker could be placed at a time per player and would last only for eight seconds apiece. The ping would usually be placed in the centre of the player's field of view. (Awyman13 et al., 2022)

Figure 1: Basic pin displayed, from the game “Rainbow Six Siege”

In “Apex Legends”, the player gets to use a whole ping wheel to choose the best fitting ping type. Each ping triggers a voice line hearable for other characters in the game and places an icon on the game map visible to all other team members. The basic ping is used to highlight items, map features or enemy ability usage along with specific ping types to inform about enemies, loot or for strategy purposes. (Jnthndrn et al., 2022) “Apex Legends” is one of the few games that have optimised this interaction to an extent, where players can communicate solely through the ping system by providing a whole environment of ping interactions instead of just a visible marker like in “Rainbow Six Siege”. (Bariş Tekin, 2021) Interestingly, as ping is a widely used feature in games and as soon as a game misses this feature or if bugs occur, many forum posts can be found in which the problem of a missing ping system is discussed and demanded. At the same time, from some players, the optional abandoning of markers, especially map markers or the user interface in general regarding role-play and open-world games is demanded. Objective markers would lessen the need for exploration and take game immersion while the player would only need a general direction on where or to which area to go. (Cameron Corliss, 2022)

If a pin or marker has been set, there needs to be a way to remove it again. To remove a shared pin there are a variety of options on how to implement this. In our case, the VR player has the sole permission to delete a shared ping. Further options would be to allow the creator of the ping to delete his ping as well or to add a timer so that the ping destroys itself after a certain time.

Figure 2: Ping wheel to select specific pings, from the game "Apex Legends"

In many multiplayer games, a pin will be removed after a certain time or if a new ping by the same user was set. In games, where a pin or marker is set on a map, most commonly the marker can only be removed by the creator, either by clicking on it or by placing a new one. The implementation of the removal strongly depends on what use case the feature is used.

A clear benefit of the ping feature is to communicate with other players without the need for voice. By providing predefined patterns, like an icon paired with a text chat output or voice line by the in-game character, which allows fully non-verbal communication, the players can avoid accidentally giving information about their intentions to other people. A further benefit is, that in comparison to an in-game voice chat, the game developers do not have to moderate any communication, as all possible communication is predefined. (Bariş Tekin, 2021)

There are not a lot of disadvantages to a well-implemented ping feature. However, a ping or marker could reduce the immersion of a game as it is a visual object that does not directly belong in the game world's context. (Cameron Corliss, 2022) Another way a user may have a bad experience with the pin feature is, if other players spam their pins, for example, to obstruct the view of a team member or to spam a triggered voice line or text chat output.

Based on the wide use and the small disadvantages of this feature, we decided to implement a version of our own. For a detailed view of how this feature is implemented, see chapter "3.7.1. Implemented Interaction 1 - Ping-Tap"

2.3.3. Implemented Interaction 2 - Pointer Stick

The idea of using a pointer stick as an extension of the index finger to manually point at presented objects probably dates back to even before antiquity. We first came in touch with this concept in school, as teachers used to have these long, finished off wooden sticks to better point at elements on the chalkboard or presented by an overhead projector. (Bibliographisches Institut GmbH, n.d.)

Figure 3: Robert McNamara pointing at a Vietnam map using a pointing stick

The simple benefit of a pointer stick is, that with this extension one can better present about what object he is talking. For example, in a presentation it is extremely helpful that the presenter does not have to stretch and reach into the projection but can easily and without much unnecessary movement point at the presentation.

While the pointer stick is still part of the technical equipment of many presentation rooms (Lehrerinnenfortbildung Baden-Württemberg ZSL, n.d.), in the age of digitalisation it gets mostly replaced by the laser pointer and is getting out of date.

As this concept is so simple and all users should be familiar with it, we decided to implement a pointer stick for the presentator, the VR host.

2.3.4. Implemented Interaction 3 - Visible AR-Users

The concept to visualise all users within the same game room, for example by giving them a virtual avatar or by displaying them as a moving object, in our case the mobile devices, is a common practice in most virtual collaborative use cases. In multiplayer games, it is usually given to see other players as modelled characters, but for example in applications for virtual meetings, visible team members have a much higher impact.

Using a player avatar can often create a more intense and satisfying game experience and may even encourage loyalty among players. Providing the player with an avatar, often with a whole character editor, allows a sometimes strong, deep and complex player-avatar connection by which players may see their avatar as a representation of their ideal selves. There are different types of connections players can establish with their avatars. Especially in virtual reality, players usually feel a strong embodiment with the avatar, as in the VR view, arms and hands are perfectly in sync with the real world. But also by controlling and configuring skills and special abilities of the character promotes embodiment and lets the player identify better with his avatar. While it would seem logical giving the avatar a heartbeat and audible breathing to support a stronger connection, quite the opposite seems to happen. If the avatar's reaction is stronger than the reaction of the player, many players seem to have bigger difficulties to feel like they are inside the game, as they cannot connect as well to the character they are playing as if they could only hear their own, real heartbeat. (Hugo Aranzaes, 2021)
A further big benefit of using avatars is the feeling of being together with another person in virtual space, the so-called co-presence. Having an online meeting and being alone can easily feel disturbing for users, whereas seeing other virtual characters, ideally personalised according to the real person, establishes a sense of co-presence and allows better experiences. According to J. Casanueva and E. Blake, who described the effects of avatars on co-presence in a collaborative virtual environment in 2001, the co-presence generated by virtual avatars differs significantly by the degree of realism the avatars resembled. Very realistic and human-like avatars would generate a significantly higher co-presence than cartoon-like and unrealistic avatars. The experiments also showed that providing simple gestures and facial expressions to avatars would further provide significantly higher co-presence than static avatars. (SAICSIT 2001 et al., 2001)
Using virtual characters may also help to strengthen the connection between co-workers or team members, increase the user's engagement, promote self-expression and inclusion and build up trust. (Danny Stefanic, n.d.)

While avatars can bring a lot of benefits, the virtual representation of a real life person can bring some risks with it. The representing avatar may not show the same credibility as the live presenter, especially if it is a respected person or an acknowledged expert. Especially unrealistic or cartoonish avatars can negatively influence and can be a turn off for some viewers. Therefore, the avatar for the presenter should be chosen very carefully, and if the viewers can also choose an avatar, it could be useful to limit the chooseable and configurable options to a serious selection. (Ken Desson, n.d.) Another interesting negative effect of the use of avatars in video games and virtual worlds was shown in a study by Jorge Peña, Jeffrey T. Hanckock and Nicholas A. Merola in 2009 which shows the surreptitiously and negatively effect on user's thoughts. By manipulating the avatar's appearance, the probability of people thinking and behaving in a predictable way could be augmented without raising suspicion. (Jorge Peña et al., 2009) It could therefore be possible to somewhat direct the virtual encounter and manipulate for example the outcome of a virtual meeting.

Based on the many benefits of this concept, we decided to implement a basic and strongly simplified avatar for the VR user and to use a moveable object to visualise the AR mobile device. It would be an interesting approach to try and use 2D virtual avatars for AR users by accessing the mobile devices camera, but that approach would be out of scope and would need to be researched independently.

3. Development of the AR-Extension

In this section, the development process, including all major design decisions and the technical implementation of both the VR as well as the AR application, is described.

3.1. Requirements and Division of Work

The project was set up by the use-case, that the model of a building with placed solar panels gets created by an already existing web application and that this building will then be presented by a VR host. The AR application then allows AR viewers to join the host and observe the model and all interactions the host does with this model. VRLabs and the AR extension will be used in an educational environment, for example, the teacher uses the VR headset and all students can observe what the teacher presents, through AR. Students furthermore will have the functionality to interact with the teacher in some way, described in theory in section "2.3. Interaction Concepts" and as technical implementation in a later section "3.7 Implemented Interaction Concepts".

The customer stated that the already existing VRLab application should get an extension for mobile AR users, so that they can observe and interact, and that the VR application should gain more realism and a more immersive VR experience.

To realise the required AR extension, two projects were built: One as the mocked VRLab application, as the already existing VRLab is not yet running on Unity Engine, and one for the new AR-Viewer. They will be referred to as AR-Viewer and VR-Player respectively.

The extension on the VR side for improved immersion and realism as well as extended functionality is not part of this project, therefore the VR-Player application will be kept as small and easy to integrate as possible.

3.2. VR-Mock-Application

As the technical implementation of this project will be integrated into the ported VRLab application, the VR-mock will be designed as easy and fast to integrate as possible.

3.2.1. Basic Structure

The VR-mock application consists of two scenes, "Launcher" and "VRLabs".
The "Launcher" scene provides the main menu where the VR user can create a new room. The rooms are given verbose names, predefined in a newline-separated file of names. This way, the names can be easily replaced, if the customer desires to do so. The user can also select an asset that he wants to use as the building. New assets can be made available by creating assetbundles. The scene consists of a basic environment, an event system used for UI, a player prefab that contains the XR rig and a prefab consisting of the UI and the launcher script. Prefabs were used to improve the integration time. The launcher script handles functionality to connect to the photon network, like joining a lobby and creating a room as well as several PUN-specific callbacks.
The "VRLabs" scene is the game room itself, where the user can see and interact with the building model. This scene consists of a simple environment, an event system for UI, a player

prefab and a prefab containing the UI as well as the game manager script. The game manager script handles the joining and leaving of the game room and instantiates the building and all necessary synchronised objects.
For a more detailed description refer to the “Integration Guide” which can be found in the appendix.

3.2.2. Basic Interaction

The basic implemented VR interaction derives from the XR interaction toolkit and allows movement by teleportation, head and hand movement and basic grab interaction.
These interactions are as close as possible to the existing implementation of the VRLabs but also as simple as possible to not lose time with the mock implementation.

Configuration Boards

All the UI and directly connected scripts are saved as a prefab that allows for it to be dropped into the scene with minimal setup needed. These UIs come with a default model, which looks like an old-school blackboard. This model can be easily replaced with a more desirable one.

3.3. AR-Viewer

The implementation of the AR-Viewer application is the main goal of this project. It includes all implementation efforts, such as the VR-Viewer extension, at runtime exchangeable models and user interactions to aid in communication between VR and AR users.

3.3.1. Basic Structure

The AR-Viewer application consists of two scenes, “Launcher” and “VRLabs”.
The “Launcher” scene provides the main menu where the AR user can see all existing rooms with the room name and the number of users per room. The user can enter a nickname and join one of the listed rooms. The scene consists of an event system used for UI interactions, a canvas containing the UI, a launcher script and a device manager prefab. The launcher script handles functionality to connect to the photon network, join a lobby and join open rooms as well as providing data for the UI room list and handling several PUN specific callbacks. The device manager script prefab handles the device orientation and the functionality of the device's back button.

The “VRLabs” scene is the game room itself where the user can observe the building model in AR and interact with the VR-host and other AR users. The scene consists of an AR session and an AR session origin which provide functionality for marker-based AR. There is also an event system used for UI interaction, a canvas containing the UI, a photon manager prefab, a device manager prefab and a VR space prefab, bundled into one, as VRLabs_ARExtension prefab. The photon manager prefab contains the photon manager script, which handles PUN specific methods and callbacks like joining and leaving a room. The device manager prefab handles, as described in the previous paragraph, the device orientation and the functionality of the device’s back button, in this case to leave the game room correctly and return to the “Launcher” scene. The VR space prefab contains all objects synchronised by PUN and equates to the global space of the VR environment. It is also used to facilitate rotation and

scaling of the AR scene. The instantiation of the synchronised objects is handled by the instantiation manager script, included in the VR space prefab.
For a more detailed description refer to the “Integration Guide” which can be found in the appendix.

3.3.2. Basic Interaction

The basic interaction in AR deriving from ARCore is basically looking around by moving the device. To enhance the user experience, a touch-gesture-based interaction allows to scale the VRSpace, while a UI element, the so-called “compass”-feature allows the user to rotate the VRSpace. With these basic interactions, the user will be able to position himself at any place in the game world.

Scaling

The scaling feature is implemented by checking if the user has two fingers placed on the screen and by then calculating the difference between the two points of the fingers. This difference is used to determine the amount and velocity of the scaling.

Rotating

The rotating feature, in this project often referred to as the compass feature, is implemented as a UI element that can be dragged. The idea is to have a UI element with the shape and look of a compass to give the user the experience of physically rotating the VRSpace. The implementation of the script is simple. It is checked if the user has one finger on the compass, and then the difference between the starting point and the endpoint is calculated to determine by how much the VRSpace should rotate.

3.4. Photon Unity Networking

Photon Unity Networking - short PUN - is the base layer for this multiplayer application and provides us with a stable and widely used networking solution. Based on the Photon engine, PUN is a tight Unity integration that further provides us with great documentation for this project. (Exit Games, 2022c) The theory of PUN is described in detail in the previous section “2.1. Photon Unity Networking”.

3.4.1. Learning PUN

PUN delivers core functionality for our project, but learning how PUN works took us quite a lot of trial and error. We first set up the joining of rooms according to the official PUN tutorial. (Exit Games, 2022a) We then created a world for VR with some environmental features to test the VR setup. In this scene different components that are provided by PUN, like the Photon View, Photon Transform View and Photon Rigidbody View were tried out. It took us some playing around to gain a basic understanding. At first, we did not understand that the PUN system does not care about what type of gameobject, or what non-PUN components are used. Key to this understanding was that at one point we had gameobjects with different models, using the same View ID. So one of us had a pile of 20cm cubes on the floor, while the other had a bunch of small 5cm cubes flying in a row, as if they had an invisible 20cm collider around them.

We realised what the components really do. Which, to be honest, is quite simple.

- Photon View:
    - Provides identification, through a View ID, of an object and serves as a hub for all PUN components. Without this component, PUN knows nothing of the object.
- Photon Transform View
    - Synchronises the selection of following variables
        - Transform.position
        - Transform.rotation
        - Transform.scale
    - A checkbox allows for the decision, if local or global values should be used
- Photon Rigidbody View
    - Synchronises a selection of rigidbody variables like velocity

Once we understood that PUN does not do any magic, better progress has been made and sources of errors were identified more easily.

The next hurdle was figuring out how objects could move, scale and position while maintaining the correct relation to each other. Of course, the easiest would have been to shift, rotate and scale the whole virtual world in the AR application. But the AR framework does not allow this, so a workaround was needed. We decided to use a single, empty gameobject called "VRSpace" as a representation of the VR environment. The local position, rotation and scale in this "VRSpace" would map 1:1 to the global position, rotation and scale of the VR environment.

Since none of our objects contained any rigidbodies for physics simulation, we realised that many of our objects would only need a Photon View component for instantiation and deletion of the object. Only very few objects, like the PointerStick and the representation of other users, would need a Photon Transform View.

3.5. Dynamic Unity Assets

The overarching project of VRLabs is concerned with different solar panel configurations on different buildings. To support this goal, we implemented a system that allows the exchange of the building models without needing to deploy the application anew. To this end, we store the models as assetbundles on a web server, where we can access and download them. Here we discuss how this is done:

We added a UI to select an asset. The list of available assets could be dynamically loaded from the server. Once an asset is selected, we download it asynchronously using LoadAssetFromWeb(), a method within the GameManager which can run as a coroutine. This method makes use of the UnityWebRequest class. The downloaded assetbundle is then unpacked and the model is instantiated locally. We also send an event to all other clients in the room, containing the name, position and rotation of the new object. This triggers all other clients to also download the assetbundle and instantiate the model.

The creation of these assetbundles requires a unity project with the same modules installed as the main projects have. We also need to make different assetbundles for AR and VR, using corresponding build settings, as they use different rendering methods. Not doing so can lead to unforeseen effects like rendering the model only on one eye in VR. We decided to make a minimalistic standalone unity project to build these assetbundles. To our knowledge, it is not possible to build assetbundles without installing Unity and all its corresponding modules. For a step-by-step guide on building assetbundles, please refer to the "Integration Guide" in the appendix.

3.6. Design Decisions

Throughout this project, we encountered major problems that would need a decision that would influence the design and base structure of the application. In this section, we will describe and explain why each major decision was made.

3.6.1. ARCore vs WebAR

We needed to decide between an application that needs to be downloaded and a web-based AR experience. The WebAR approach would have the advantage of easy access. We did some research into developing multi user WebAR applications through Unity, and found no solutions. There are some projects that aim to implement this possibility, but the ones we found were abandoned or would not support major features that we would need. Thus, we decided to build an application using ARCore, which is well supported in Unity.

3.6.2. Marker-Based vs. Markerless AR

Whenever we have mobile AR, we need to decide whether we want to use markers or not. Markers allow for easy positioning and repositioning of the model by simply moving the physical marker. Furthermore, according to Jack Cheng, Keyu Chen and Weiwei Chen, the marker based AR has a significantly higher accuracy, but a lower stability. (Cheng et al., 2017) Since instability is, at worst, a source of slight annoyance, we decided to go with the more commonly used method of marker based AR. Nonetheless, the application can be used without a marker, at the cost of intentional positioning of the viewed objects.

3.6.3. Unity Networking with Photon

Since we need to synchronise many game objects between the VR player and the AR viewers, we needed some kind of networking framework. During research we found some applications that were quite close to what we wanted to implement. They used PUN. Upon further research into PUN, we found that it had a wide user base, good reputation and excellent documentation.

3.7. Implemented Interaction Concepts

Three interaction concepts were implemented, not counting the visualising of the VR player.

3.7.1. Implemented Interaction 1 - Ping-Tap

The Ping-Tap interaction, or simply Ping, allows the spatial marking of scene features from AR to VR.

Concept

The idea is quite simple: the AR viewer uses the touch functionality of their mobile phone to select a desired location on the shared objects, in our case buildings, to place a pin. He can then select to send this pin to all other users, including the VR player. This shared pin can now be deleted by the VR player, but not by any of the AR viewers, including the original creator.

Implementation

There are currently two implementations of this interaction in our project. Our original one, and one that was resulted from feedback.

Deletion

Both versions use the same VR-side logic for deletion. We added an XRSimpleInteractible to the pin object. This allows for the registration of methods for different callbacks. To visualise the possibility of deleting the pin by pointing at it, a bin symbol gets displayed above it when the VR player points at it with their hand. Deletion happens once the VR player presses the grip button while pointing at the pin. Since the object was created by someone other than the VR Player, we cannot delete it directly. To delete it, a special event, called a remote procedure call (RPC), is sent to all other users. If the user owns the pin, he triggers the deletion of the object.

Ping Version 1

This is our first version of the ping. It uses a long-press to set the pin, which we chose to prevent accidental setting or moving of the chosen location. Specifically, the implementation works as follows:
On every update, we check if we are in ping mode and if there is exactly one touchpoint on the screen. If so, and if the touch has only just begun (TouchPhase.Began), save our current time. On every subsequent update, where the first two conditions still hold true, we check if the difference between our current and saved time is greater than a threshold. If so, we make a raycast from the touch position, perpendicularly to the screen. Like this, we find any objects that are directly underneth the touch. We then move a pin, or create one if none exists, to that position. There is also a _onlyOneFinger variable, which we use to prevent the accidental setting of a pin when scaling the object. This is necessary because any human that tries to touch a screen with two fingers at once will accidentally touch with one finger slightly earlier. The same happens when releasing the touch. So we needed to ensure that whenever the number of touches exceeds one, we wait until we had zero touches until we check a long press again. The preview pin can then be shared with all other users with a button press.

During user tests, we found out that the long-press mechanic was not intuitive enough, therefore a second version was implemented.

Ping Version 2

In comparison to our original version, the second version now disables the ping mode by default. The user can activate the ping mode with a button press. When doing so, a preview pin will be dropped in the centre of the screen, given that there is an object in that area. This is done to make the ping action apparent. Once the ping mode is active, the user can move the pin by simply touching the screen and the pin will move automatically to any object underneath that touch point. The preview pin can still be shared with all other users with a button press.

This version should be much simpler and more intuitive to use than our initial one. Due to time constraints, we could not verify this through user tests.

3.7.2. Implemented Interaction 2 - Pointer Stick

The pointer stick is a very simple stick that allows the VR Player to point at things without having to move close to them. The idea originates from the sticks used by teachers to point at things on blackboards or projections. Today in the classroom most of those sticks have been replaced by laser pointers.

Concept

The concept is very simple:
Take an elongated object, make all users see this object and finally make the VR player able to hold this object in his hands.

Implementation

Once the general setup of our application was completed, implementing the pointer stick was very simple:
The synchronisation of the position and rotation of the stick is done using PUN. We use the global position in the VR environment, which is then mapped to a local position in AR. This is done for all objects that need to be seen in both worlds. The ability to hold the stick is provided by the XRGrabInteractible from the XR Interaction Toolkit using OpenXR.

3.7.3. Implemented Interaction 3 - Visible AR-Users

Showing the other users is important to inform them about the presence and position of other users. This should, especially for the VR Player, help in communicating with other users and enable the feeling of a co-presence.

Concept

The concept is quite simple: Show where, in relation to the building, the other users are.

Implementation

The implementation requires a bit of an understanding of the different local and global positions of objects:

In general, any object that should be shown in relation to the building and VR Player has to be a direct child of the same game object, called “VRSpace”. All objects of the VRSpace have their local coordinate mapped to the global coordinate of the VR environment. The VRSpace is also the game object that is rotated and scaled. Given this information, if we want to show the AR Viewers position in relation to the building, we need the AR Camera to be a direct child of the VRSpace, but the AR Camera is part of the AR Session Origin. To solve this dilemma, we create an empty game object “CameraTracker”, which is part of the VRSpace and attach to it a script that changes the CameraTrackers global position to the global position of the AR Camera with every update. We then synchronise the local position of the CameraTracker with the network.

3.8. Integration Guide

As the VR side of the application was mocked, we must assume that at some point, our project will be migrated to the original VRLab application. To enable a fast and easy integration process, we created a detailed step-by-step guide which describes all fundamentally necessary as well as all modifiable parts of both applications. You can find the “Integration Guide” in the appendix.

4. Usability and User Experience Testing

Usability and user experience testing allows us to gather very valuable feedback about major and minor issues with features of our application and can help to identify major pain points and structural errors within the application design.

4.1. Preparation and Planning

As user testing was done relatively late in the project, the decision was made to conduct all user tests within one afternoon. To prepare and plan, at first, all features and aspects that needed to be tested were written down. Then, two scenarios were created.
Scenario 1: The moderator will play the VR user, and the testers will focus on the AR application. This scenario will be simulating a lecture, where the teacher is leading and the students can observe and interact in AR to solve the tasks given by the teacher.
Scenario 2: The testers will play the VR application as well as the AR application. This mode will be simulating a lecture, where students have to work together to solve tasks given by the moderator.

With these two scenarios, both most likely used use-cases of our application will be recreated and will provide us with the best possible feedback.

To generate enough feedback, two groups per scenario with five testers each will be used, resulting in 20 testers in total. Each group will absolve the test in the form of a guided workshop. Each workshop should last about 45 minutes, consisting of 5 minutes warmup, 20 minutes main part, 10 minutes questionnaire and 10 minutes cooldown. To encourage honest and viable feedback, during and after the cooldown phase, some snacks and drinks will be offered to the testers.

4.2. Questionnaire

The aim of the usability and user experience test conducted in this project was to collect feedback and find the biggest pain points and structural errors in the AR application. To generate analysable feedback, the testers have to fill out a questionnaire, giving information about the demographics of the testers, giving feedback about the usability by giving scored answers as well as giving open feedback. The demographic part of the questionnaire is mainly for analysis so that we can check if our tester group fits, as well as to identify and justify possible outliers. The scored answers are inspired by the questions asked to determine the system usability scale (SUS) but adapted to give us better insight into our specific application and features. The open questions were mainly to give us better and clear formulated feedback and statements about specific parts of the application regarding the user experience as well as the usability and to clearly identify pain points.

4.3. Recruiting Testers

For the user test workshop, a total of 20 testers were planned. Of the total of 20 testers, in the end, the user tests were conducted with a total of only four testers.

Our target group are students with physical and electrical backgrounds, as the application would most likely be used in an educational environment. As the user test was conducted late in the project and therefore late in the semester, our target group had lots of exams and deadlines. Therefore, we decided to adapt our target group of testers to students in general. As it was known that most students would not have a lot of free time during this time of the semester, we advertised our workshop with a free drink and snacks. We directly addressed 30 students and reached out to around 100 students, of which only two signed up for the testing and an additional two which we could recruit during the testing day at the FHNW.

The recruited testers fit perfectly fine into our test group and cover very well all kinds of user subgroups. We have students with no to little AR/VR experience, some with much to expert level, equal number of male and female testers and one older tester.

Test Person	Gender	Age	VR Experience	AR Experience
Tester 1	Male	20-29	100+	11-100
Tester 2	Female	30-49	0	1-10
Tester 3	Female	20-29	1-10	0
Tester 4	Male	20-29	1-10	11-100

Table 1: The test group's demographics and prior experience with AR and VR

4.4. Conduction

After the testers have arrived and been welcomed and all testers have the application installed, the workshop starts with the warmup phase. The users will fill out the disclaimer and will be instructed and introduced by the moderator along the lines:
"In the last two weeks, you have learned about solar power's physical and electrical characteristics and different solar panels. As of now, your teacher gives you the opportunity to play around with different configurations of solar panels and the buildings they are seated on, to give you a better understanding of how solar power works and what influences the effectiveness of solar panels."
[Scenario 1]
"Your teacher will be playing with the VR headset, you can follow with the AR application on your mobile device."
[Scenario 2]
"Your teacher will observe the workshop, one of your group may play the VR user, and all others can follow on the mobile device. You can take turns playing the VR application."

Then, the users will start with the main part, consisting of several tasks for the VR player and the AR users to absolve. If the VR user is not played by a tester, one of the moderators will play this role.

The tasks for the VR player:

- "Open a game room."
- "Have a look around and get familiar with the game world and observe if all other students joined the room."
- "Hide somewhere behind the building, so that the AR users move around to find you."
- "You have now talked about the building, time to load the first solar panel configuration."
- "Let's say you (the VR player) wants to talk about a specific solar panel, somewhere in the middle. Find a way, so that all users of the group understand which panel is meant."
- "Load the next panel configuration."
- "Load another project/building."

The tasks for the AR user:

- "Join the game room the VR player has opened."
- "Scan the marker to position the VRSpace correctly."
- "Have a look around and get familiar with the game world and observe if all other students join the room."
- "Let's say you (the AR user) has a question or a note about a specific solar panel. Find a way, so that all users know which panel is meant."
- "As an AR user, you have the ability to see and observe all configurations and calculations of the world/project, find the following information, in detail, find:
    - Room Name (Raum),
    - Project Name (Projekt Name),
    - Energy in DC, AC (DC Energie, AC Energie),
    - Power (AC Leistung),
    - Panel Type,

- Energy production in a year (Energieproduktion in einem Jahr)."

With these tasks, each feature of both applications will have to be used. By not directly telling the testers where or which feature they need to use to complete a task, we also get feedback about how intuitive a feature is.

Following the main part, the tester will fill out the questionnaire and then discuss with the moderator all the good and bad experiences in an open discussion. In this final conversation, the moderators will try to get the feedback a tester would not be able to write down in detail, for example, what feelings the user experienced, more detailed reasons and possible future features or general improvements.

As of the planned 20 testers, only four could be recruited, each tester could try out both the AR as well as the VR application in depth after the workshop. By that, the most possible feedback could be gained.

For all workshops, scenario 1 with the moderator as the VR player was used. Group 1 consisted of one user who first completed the workshop and then tried out the VR application. Group 2 consisted of two users who first completed the workshop, one of which tested the VR application afterwards. Group 3 consisted of one user who attended the workshop remotely.

5. Results User Test

The user test results revealed the biggest pain points and structural errors as well as the strengths of the application, even if the application could only be tested against four users. This section will be divided into the notes and observations the moderator documented during the user tests, the feedback of the open questions the users provided and the analysis of the scored answers of the questionnaire.

5.1. Findings and Observations

In both applications there are two scenes, the launcher scene, which serves as a lobby to join a game room and the "VRLabs"-scene, the actual game room.

AR Launcher Scene

In the launcher scene of the AR viewer, two issues were found. The first issue is more of the nature of a bug and describes the section of the room list, which displays all currently open rooms, which will always be scrollable and always displays a scrollbar, even if there are no rooms open. The second issue is that the user cannot enter special characters as their name. For example, a tester could not enter a Japanese character as their username, either the input field would not accept the input or the character gets replaced with the "glyph not found" glyph. After closer inspection of this issue we found out that the font "Text Mesh Pro" (TMP), a UI element to display text in Unity uses, does not support all UTF-8 characters. Therefore, it would be needed to change to a different font to support further characters.
Apart from these two issues, the launcher scene was described as fast and easily understandable.

AR VRLabs First Impression

For the VRLabs scene of the AR viewer, there were several major issues identified. First, users that do not have prior AR experience found it difficult to understand, why and how they should use the marker. Then, most users had difficulties understanding the basic interactions and what functionality which button has and what device orientation is given. Though most testers could figure out by playing around, what the basic interactions are, how they work or in which orientation the application is displayed, a general need for a tutorial or introduction screen was voiced.

Scale-Feature

The basic interaction of scaling the game world was found fast and very intuitive by all users.

Rotation-Feature/Compass-Feature

The basic interaction of rotating the game world was found quickly, but was only identified as such by playing around. Most testers expressed ambiguity of the functionality at a first impression and said that celestial direction marks could help to identify the feature as a compass and give clarity to its functionality.

Reposition in VRSpace

After learning the basic interaction of scaling and rotating, all users could easily reposition them to absolve the task to find the VR player.

Identify described Panel

All users could easily identify the solar panel that was indicated by the VR player, whether he described one by hand or using the pointer stick.

Ping-Tap-Feature

The second major issue came up with the ping-tap feature. At the time of the user tests, this feature was not long ago implemented and not well tested before. Nevertheless, we want to describe the findings from the user test. First, all testers could not determine what the feature does, as the icon used for the ping mode toggle was the google material icon for location instead of pin. After a quick explanation of the feature, the testers found it very unintuitive to place and drag the ping using a long tap. Furthermore, the colour, the shape and the size of the ping were criticised, as depending on the colour of the building or surroundings, the ping could hardly be seen. The last issue found was when multiple users set pings, it cannot be differentiated who set which ping. This leads us to the need to test if it is needed to be able to tell which ping belongs to whom, if it would be a needed addition to display the username above the ping and if that would only be displayed for the VR user or all users.
Due to this feedback, the ping-tap feature was reworked shortly after.

Overlay-Feature

The overlay feature was found quickly, but regarding the intuitiveness of the navigation, half of the testers found the arrow- and carousel-based page navigation counter-intuitive, as they were more used to the android default navigation consisting of a drawer. This leaves us open to verifying both variants with a bigger tester group to identify the most fitting page navigation design.

Visible AR users

The visibility of the other users was voiced as great by all users, though an issue in the form of a bug was identified. The shader of the front and back label, which displayed the users nickname, on the mobile phone model was set wrong and therefore, both labels got displayed at the same time without being occluded by the phone itself.

In general, the overall impression of the testers was positive, but it was voiced that the application lacks further AR-to-AR interaction as well as AR-to-VR interaction.

In the table below, all identified AR viewer related issues are prioritised based on their criticality and impact.

Issue	Priority	Counteracted
Missing tutorial/introduction screen	4 - Critical	partly
Counter-intuitive ping-tap feature	4 - Critical	yes
Counter-intuitive page navigation in the overlay feature	3 - Serious	no
Faulty rendering of the mobile phone labels	2 - Medium	yes
Hardly visible ping	2 - Medium	yes
Unclear functionality of compass feature at first impression	2 - Medium	no
Visible scrollbar/scrollable at room list	1 - Low	no
Non-UTF-8 font	1 - Low	no
Unidentifiable ping	1 - Low	no

Table 2: Prioritised Issues

Due to the user testing being late in the project, many issues could not be counteracted within the duration of this project.

VR Launcher Scene

The testers that tried out the VR labs found the launcher scene fast and easy to understand. One minor issue that was found and directly counteracted was that a player could teleport behind the barrier wall.

VR VRLabs Scene

In general, the VRLabs scene was described as very cool and it was voiced that the visible AR users are very helpful. One issue identified was that it is extremely difficult to pick up the pointer stick, as the grabbable sphere is too small. Further issues found were regarding the ping-tap feature, such as unexpected behaviour of the networked pings or that the pointer stick could collide and move pins. All of the identified issues in the VR application were counteracted and resolved.

5.2. Open Feedback

The answers to the open feedback were visually prepared to give a more visual overview and to enable a bigger picture.

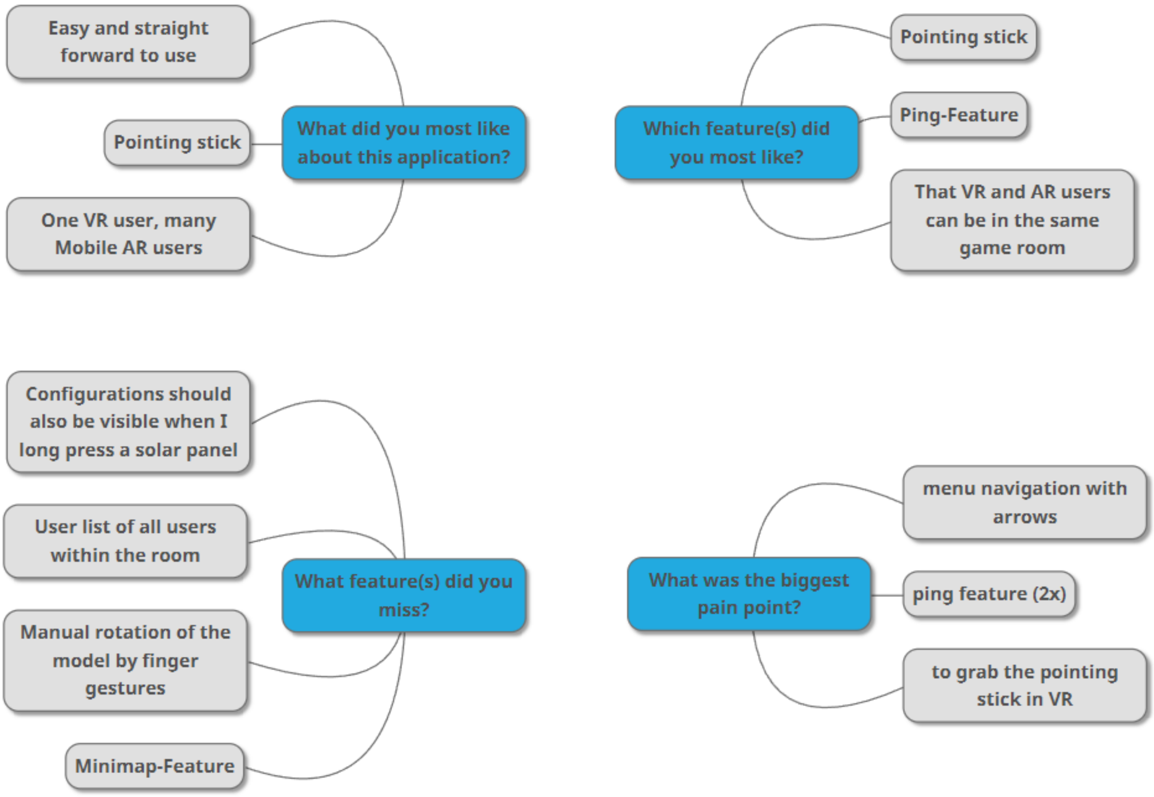


Figure 4: Overview of open feedback questions

The answers to the last question, if the tester would like to tell us anything else, were both regarding the page navigation design of the overlay feature as described in the previous chapter.

5.3. Scored Answers

As part of the questionnaire, the scored answers should provide us with a determinable score about the usability of our application. It is important to say that these values result out of only four testers, to get data which can be statistically analysed we would need more testers. The following analysis is done only to give a broad overview and should not be considered statistically accurate, as with only four testers possible outliers cannot be clearly identified.

During the questionnaire, the testers had to fill out eleven questions, of which nine questions were formulated positively and two were formulated negatively. For each question, the testers could give a score from one to five (strongly disagree to strongly agree). A score of three is therefore considered as neutral. Mathematically, the worst possible total score would be a 19 and the best possible score would be a 47, considering giving the positive questions the lowest score and the negative questions the highest. A neutral total score would therefore be 33.

In the following figure the total score per tester is being displayed, showing all testers gave a score of above neutral with an average total score of 41.

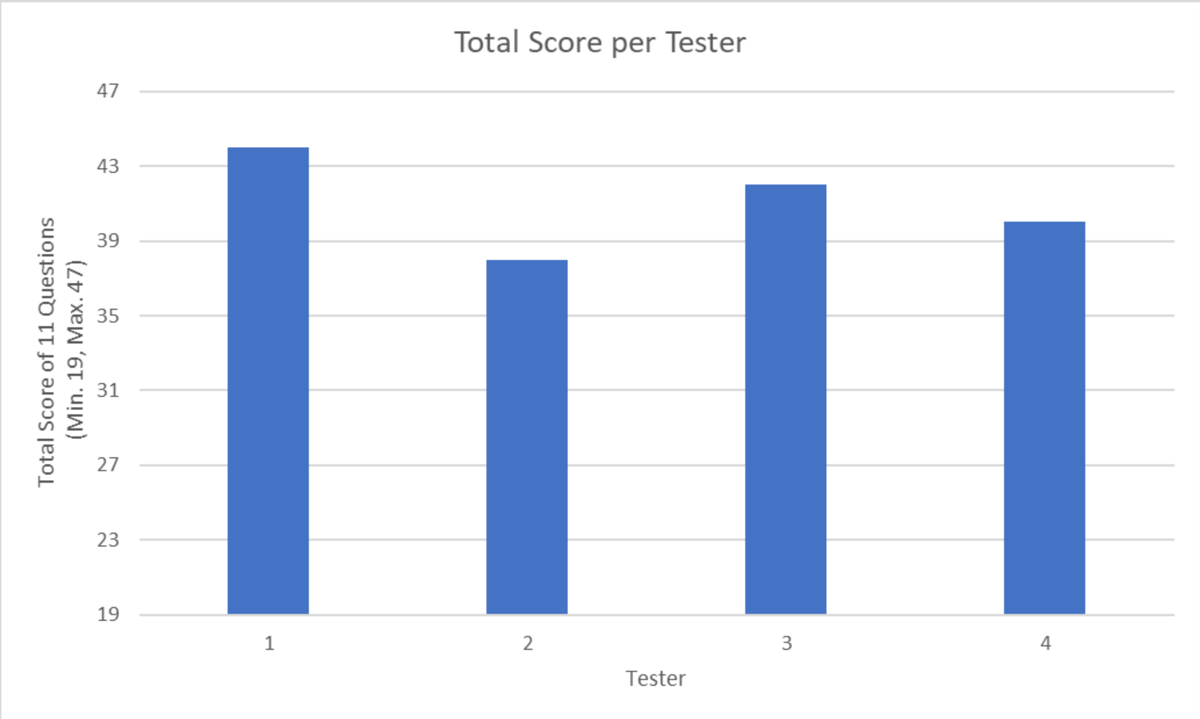


Figure 5: Total score per tester

Next, we can analyse the mean score given for each question. Each question is labelled as positive or negative question, for example is the fourth question from the left a negative question, meaning that the best score a user could give is one, whereas for positive questions the best score would be a five. For visibility purposes, the graph's y-axis starts with zero rather than the lowest possible score one, as it would look like a missing data point if the lowest score would be given. We can observe that question 1 and 2 are close to neutral, a bit above, whereas all other questions are well rated (one score above neutral).

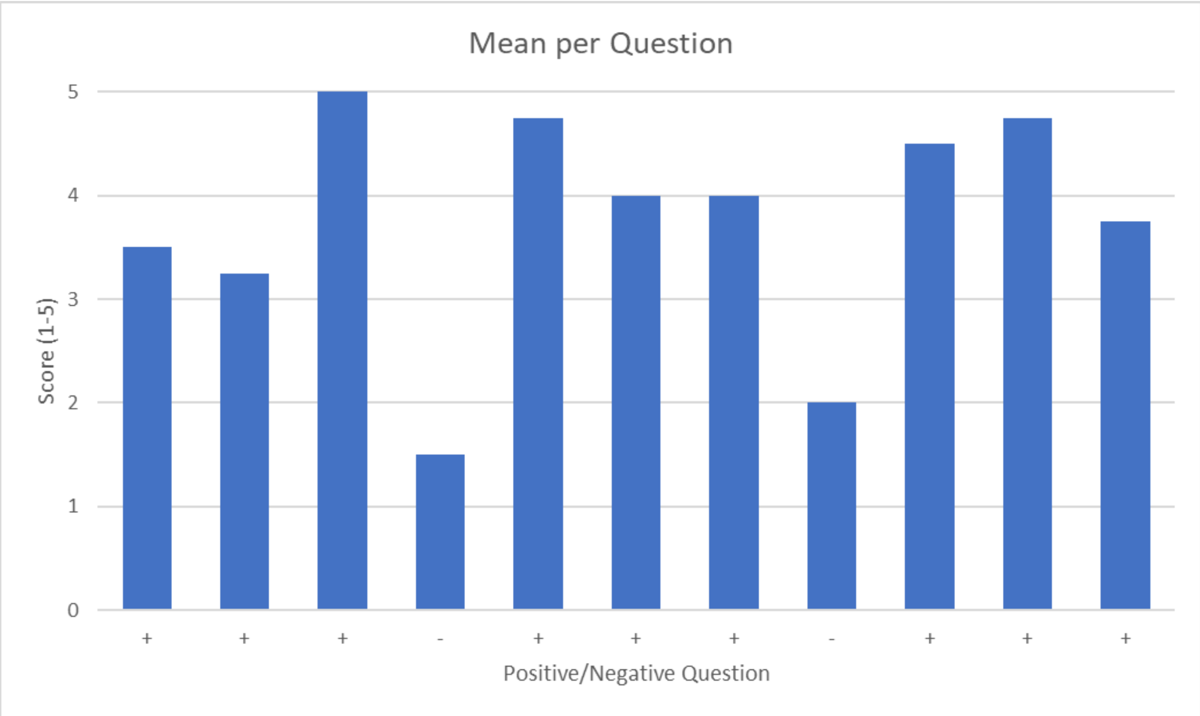


Figure 6: Mean score per question

Finally, we can analyse the score given by each tester for each question. This gives us closer insight into the score distribution within the testers. We will have a closer look into deviating values.

Question 2: This question is deviating, as half of the testers rated it below neutral, the other half above neutral. It would be interesting to see if this behaviour correlates with the previous AR/VR experience or with other demographic aspects, as both testers who gave the better score have the higher previous experience.

Question 8: This question is deviating, as half of the testers strongly agreed with the question, whereas the other half of the tester was only neutral. Again, both the well scoring testers have a higher previous experience.

Question 11: This question is deviating, as only one tester gave the worst possible score, whereas all other testers gave a high score. It would be interesting to see if this data point could be considered as an outlier.

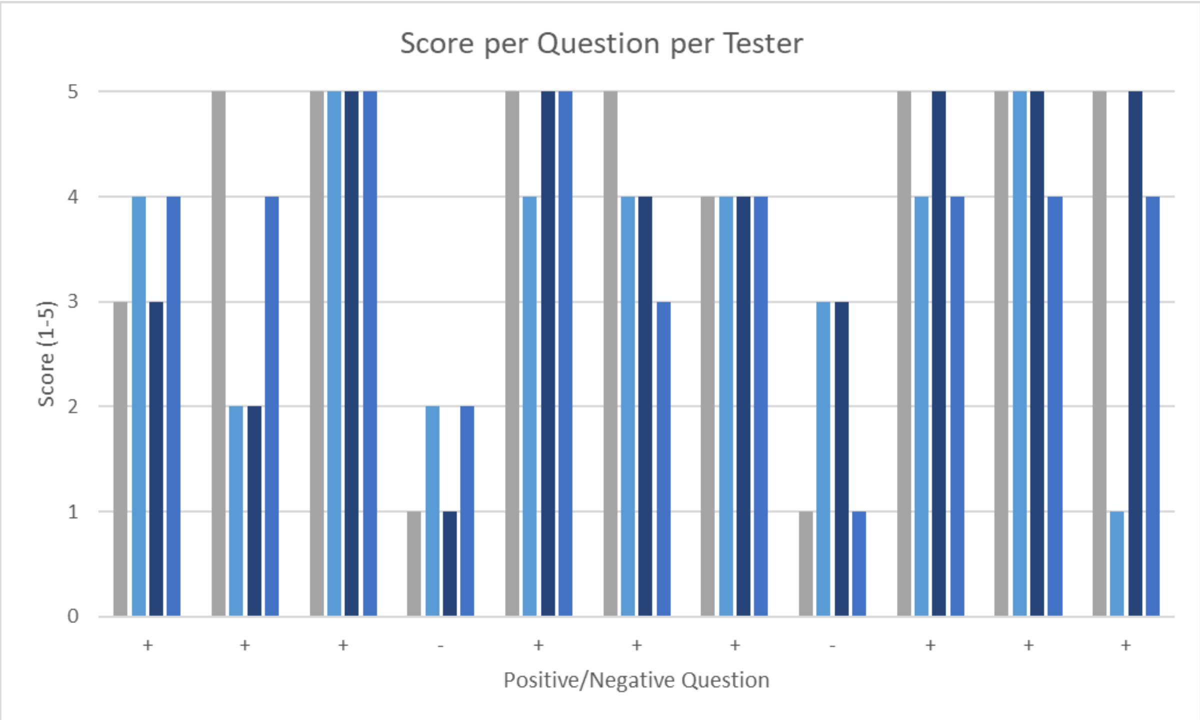


Figure 7: Score per question per tester

With the planned amount of 20 testers, it would be interesting to see if these statistics would behave the same way or whether the testers would continue to disagree with each other.

6. Discussion

This section elaborates on the meaning, importance and relevance of our achievements and results of this project as well as possible future extensions and a few final words.

6.1. Summary Technical Achievements

In this project, an AR extension to the VRLabs was implemented. Due to the ongoing migration of the main VRLabs application from Godot to Unity, a mock VR application was also realised. The VR application was kept as simple as possible and is constructed in a way that should allow an easy integration.

For VR, OpenXR and the XR interaction toolkit were used. For AR, Android's ARCore was used. PUN was used to facilitate the synchronisation of elements across the different applications. With the help of PUN it is now possible to see defined objects from the VR application in an AR environment, while allowing for scaling, moving and rotating the whole scene in AR space. Different tools that support communication and cooperation between the user in VR and the user in AR have been devised and implemented. By this, the objective of extending the VRLabs by a mobile AR viewer as well as the objective of implementing and user testing interaction concepts have been achieved. With PUN, a technical concept was used to achieve a synchronised solution for AR and VR users.

Using Unity assetbundles, the application allows for the exchange of models, specifically of buildings in the VRLabs, without the need for a new deployment of the app. The app could also easily be modified to allow for model exchange during runtime. With this, the third main objective is achieved. We proved and implemented a way to modify, exchange and synchronise unity assets at runtime.

6.2. Summary User Study

The AR application was tested with only four persons. As the test was conducted late in the project and therefore late in the semester, our target group had lots of exams and deadlines. The test was conducted in a task-based usability workshop with a subsequent questionnaire, which revealed the biggest pain points and structural errors as well as the strengths of the application.

During the usability test, many issues have been identified to later be prioritised and counteracted. Both testers with and without prior AR and VR experience could easily learn to use the application. The two most critical issues found were that users have had difficulties understanding all basic interactions or the intended device orientation and that the first version of the ping-tap was very difficult and unintuitive to use. Both of these issues were immediately addressed, once by implementing a tutorial screen and once by reworking the ping-tap feature, resulting in a second version to fully ensure the correctness, usability and reliability of all our features.

A further interesting result of the user study was the disagreement between the testers, how the page navigation of the overlay feature should be implemented. For us, both the current arrow-based navigation as well as the drawer-based navigation would make sense. This is one of the issues that would need the planned 20 testers to clarify.

The general feedback was positive. Users liked the way VR and AR players could be in the same virtual room and that AR devices are visible to each other. Each tester acted and solved each task a bit differently and it was very important to observe, especially the testers first impression and first steps with the application as this gives great insight about intuitiveness.

In general, the AR-Viewer is a great support to the VR application by allowing interaction and collaboration between all users.

6.3. Project Management

As planned, this project was managed in an agile style, allowing fast decisions and frequent changes. On bi-weekly meetings the current state of the development was demonstrated, reviewed and together with the customer as well as the advisors discussed. All parties could add their ideas, wishes and concerns which lead us on a fast and close collaboration and gave room to great possible future extensions.

6.4. Challenges

This project's main focus was on Unity, VR- and AR-development as well as the use of PUN. For us, the biggest challenge in this project was to get to know PUN, how it works and how it can be used. We spent a considerable part of time solely on understanding PUN. Further, both of us had no prior experience in AR development. The whole process of how image recognition and tracking works, how objects are bound to this image and how to use Android’s ARCore was a big challenge to overcome. Last but not least, for Thierry, a further challenge was that he did not have any prior experience in VR development and only very low experience with Unity itself.

Apart from having to research and learn every framework and technology used in this project, one of the biggest challenges was to properly user test our application. We were only able to start a user study workshop late in the project and did not expect as much difficulties in finding available testers as we had.

6.5. Comparison to Original Project Description

In the original project description, the bigger focus was on the implementation and useability testing of interaction concepts. This focus was reduced to only integrate a few interaction concepts and elaborate further concepts theoretically due to a customer decision to add the functionality to load and change game assets dynamically.

Further, the application was only developed for Android mobile AR rather than also for cardboard VR due to a customer decision.

In total, the initial objective of the original project description could be achieved.

6.6. Possible Future Extensions

There are many more possible extensions to the interactions, the AR extension, as well as to the VRLabs application.

Interaction Concepts

Here we discuss a non-exhaustive list of possible future additions to the interaction concepts.

Virtual Whiteboard

One possible extension would be a writeable, virtual whiteboard that is synchronised to a 2D UI for the AR viewer. This would allow for simple and easy sketches and notes and would make them easily readable, as the AR viewers would not need to look at them in the 3D context of the world. See a detailed theoretical approach in chapter "3.2.1. AR-VR Interactions: Virtual Whiteboard".

Highlight-Feature

Another possible extension would be to allow the VR player to highlight any surface in the game world, for example using a spray can or a big marker. This would allow fast and simple notes directly in the virtual world and would somewhat define the counter interaction to the ping-tap feature. See a detailed theoretical approach in chapter "3.2.1. AR-VR Interactions: Highlight-Feature".

Polls

Allowing for polls or similar interactions might be a helpful feature to reach decisions quickly. See a detailed theoretical approach in chapter "3.2.1. AR-VR Interactions: Polls".

Labelled Markers

For some cases it might be useful to place markers that contain a text at specific places. This would be especially useful with another extension, the Breakout Rooms or Standalone AR Rooms. With these, it could be used to prepare configurations for studying outside of the general class. One could also add larger text and image notes to these pins that could be called by tapping on them.

AR Extensions

Here we discuss a non-exhaustive list of possible future additions to the AR extension part of the VRLabs.

Rotate by gesture

An alternative method to rotate the world would be by gesture, similar to the zoom feature.

Measurement Tool

Since there is no real reference point in the AR environment, it might be helpful to add a measurement tool to measure distances in the virtual world.

Breakout Rooms

For certain situations in class, it might be helpful to split into smaller groups to discuss a certain configuration. Combined with the Editing Configurations in AR, this would allow for splitting up in groups to try and optimise a configuration. With the correct VR Metaphor implemented, the VR Player could then display one configuration after another and discuss their strengths and weaknesses.

Editing Configuration Variables

Editing some of the variables of a configuration in AR might be useful. Especially if any of the AR Viewers wants to try out a certain configuration, it would reduce the time spent communicating which variable should have which value. This would also be helpful for Breakout Rooms.

Standalone AR Rooms

Currently, there always needs to be a VR Player present to study a configuration. If we allow AR Viewers to open their own rooms, they could study outside of class and without the headset. With enough other features implemented one could even provide materials for self study by adding notes to areas and allowing the VR Viewer to edit the configuration.

VRLabs Extensions

Here we discuss a non-exhaustive list of possible future additions to the VR part of the VRLabs.

Standalone VR Hardware

Currently using the VRLabs requires the Oculus Rift, a powerful enough computer and controllers. Setting this up has quite an overhead and the Oculus Rift is cablebound, which can be inconvenient and makes the user immobile. By porting the project to a standalone headset, like the Oculus Quest, the overhead could be reduced and the mobility massively increased.

Hand Tracking

Extending on the above mentioned point of standalone hardware, implementing support for hand tracking would further reduce the overhead and would increase the immersion. We do not recommend this change without the respective adaptation of the VR Metaphors.

Realistic Environment

To make the experience more immersive, a more realistic environment would be helpful. It is important that this environment does not impair the functionality. As an example, it would be detrimental to the experience if there was a tall mountain in the environment, but the mountain would not be part of the panel simulation.
Furthermore, one could add a realistic, real world environment that factors into the calculations. This would require well prepared data, as we do not want to have our building clipping with the environment, or the environment delivering wrong information about the simulated situation.
Having a relevant environment might also prove a challenge for the AR extension to visualise.

VR Metaphors

The current VRLabs uses very basic movement metaphors and relies on flat-screen UI interactions to edit configurations. This makes very little use of the advantages of VR and decreases immersion. Examples of better metaphors would be:

- Replacement of dropdown menus with a rack filled with the selectable components, allowing the user to grab and place these objects in a 3D configuration
- Replacing 2D Sliders with 3D objects that can be moved
- Removing the daytime slider and setting the time according to the sun, which can be dragged across the sky

Chat Tool Integration

While we do not believe that the application needs its dedicated chat functionality as this is better provided by existing external solutions, such as MS Teams or similar, we do believe that an integration of such a tool could be beneficial. VR applications tend to make accessing things outside of the application cumbersome. Thus, adding the ability to read text chat or see an ongoing meeting from inside the VR application might be helpful.

Measurement Tool

As in AR, there is not currently any easy size reference in VR, thus it might also prove beneficial to add a measurement tool to VR. One could also consider providing the ability to share these measurements with the other users.

6.7. Final Words

For both of us, this project was very interesting and offered us a great learning experience. For Andreas, it gave a further great base for his future career in game design and development, as experience with Unity is extremely useful in this field. For Thierry, the personal goal of learning Unity and getting to know AR and VR development was also achieved and levels the path for future projects using Unity and augmented reality.

Both of us have great ideas of how to further improve both the AR and the VR application and to further strengthen the user experience and hope to see future development of this project.

7. Acknowledgements

First of all, we would like to thank Renato for making this project happen and giving us the chance to dive deep into the development of virtual and augmented reality. We would also like to thank Arzu and Cloe for their great support during this project. It was always pleasant to work with you, especially due to the open and relaxed interpersonal atmosphere.

We would also like to mention and thank the four testers that found time to thoroughly test our application and give us valuable feedback.

8. References

Awyman13 et al. (2022, April 28). *Ping—Rainbow Six Wiki* [Community content, Wiki]. Ping | Rainbow Six Wiki | Fandom. https://rainbowsix.fandom.com/wiki/Ping

Bariş Tekin. (2021, November 3). *Apex Legends 101 Chapter 7: The great ping system of Apex Legends*. Apex Legends 101 Chapter 7: The Great Ping System of Apex Legends – G-Loot. https://gloot.com/blog/apex-legends-101-chapter-7-the-great-ping-system-of-apex-legends

Bibliographisches Institut GmbH. (n.d.). *Duden | Zeigestock*. Duden | Zeigestock | Rechtschreibung, Bedeutung, Definition, Herkunft. Retrieved 16 June 2022, from https://www.duden.de/rechtschreibung/Zeigestock

Bryan Wirtz. (2021, June 24). *What You Need To Know About Photon For Multiplayer Games (And Where To Find Useful Tutorials, Helpful Comments, and More)*. Photon Engine For Multiplayer Games: Try It for Free (What You Need To Know). https://www.gamedesigning.org/engines/photon-multiplayer/

Cameron Corliss. (2022, January 26). *More Games Should Ditch Objective Markers*. More Games Should Ditch Objective Markers. https://gamerant.com/objective-markers-ditch/

Cheng, J., Chen, K., & Chen, W. (2017). *Comparison of marker-based AR and markerless AR: A case study on indoor decoration system*. *2*. https://doi.org/10.24928/JC3-2017/0231

Danny Stefanic. (n.d.). *The Top 5 Evidence-based Benefits of Using Avatars for Your Next Virtual Event*. The Top 5 Evidence-Based Benefits of Using Avatars for Your Next Virtual Event - MootUp. Retrieved 16 June 2022, from https://mootup.com/the-top-five-evidence-based-benefits-of-using-avatars-for-your-next-virtual-event/

Digital Extremes Ltd. (n.d.). *Warframe: Ninjas Play Free*. Retrieved 13 June 2022, from https://www.warframe.com/de/landing

Discord. (n.d.). *Discord*. Discord | Ein Ort Zum Treffen Und Zum Unterhalten. Retrieved 13

June 2022, from https://discord.com/

Ducheneaut, N., & Moore, R. J. (2004). The social side of gaming: A study of interaction patterns in a massively multiplayer online game. *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work - CSCW '04*, 360. https://doi.org/10.1145/1031607.1031667

Electronic Arts Inc. (n.d.). *Apex Legends*. Apex Legends – Die Nächste Evolutionsstufe Des Helden-Shooters – Free-to-Play. Retrieved 16 June 2022, from https://www.ea.com/de-de/games/apex-legends

Emily R. (2022, May 5). *In-Game Chat 101—Text, Voice, & Video Messaging*. In-Game Chat: An Intro to Text, Voice, & Video Messaging. https://getstream.io/blog/in-game-chat/

Exit Games. (2022a, May 6). *0—Introduction* | *Photon Engine* [Tutorial]. PUN Basics Tutorial. https://doc.photonengine.com/en-us/pun/v2/demos-and-tutorials/pun-basics-tutorial/intro

Exit Games. (2022b, May 6). *Introduction* | *Photon Engine*. Introduction. https://doc.photonengine.com/en-us/pun/current/getting-started/pun-intro

Exit Games. (2022c, May 6). *Multiplayer Game Development Made Easy* | *Photon Engine*. Multiplayer Game Development Made Easy. https://www.photonengine.com/en-US/Photon

Exit Games. (2022d, May 6). *Optimization Tips* | *Photon Engine*. Optimization Tips. https://doc.photonengine.com/en-us/pun/current/gameplay/optimization

Exit Games. (2022e, May 6). *Ownership & Control* | *Photon Engine*. Ownership & Control. https://doc.photonengine.com/en-us/pun/current/gameplay/ownershipandcontrol#ownership_transfer_options

Hugo Aranzaes. (2021, January 22). *Using player-avatar relations to make more engaging and successful games* | *GamesIndustry.biz*. https://www.gamesindustry.biz/articles/2021-01-22-using-player-avatar-relations-to-make-more-engaging-and-successful-games

Jnthndrn et al. (2022, March 19). *Ping—Apex Legends* [Community content, Wiki]. Ping - Apex Legends Wiki. https://apexlegends.fandom.com/wiki/Ping

Jorge Peña, Jeffrey T. Hanckock, & Nicholas A. Merola. (2009, November 11). *Avatars Can Surreptitiously And Negatively Affect User In Video Games, Virtual Worlds*. Avatars Can Surreptitiously And Negatively Affect User In Video Games, Virtual Worlds -- Science Daily. https://www.sciencedaily.com/releases/2009/11/091110211037.htm

Kayaba. (n.d.). *SWG Legends*. SWG Legends | Home. Retrieved 13 June 2022, from https://swglegends.com/

Ken Desson. (n.d.). *The Pros and Cons of Avatars as On-Screen Mentors*. The Pros and Cons of Avatars as On-Screen Mentors | Androcom. Retrieved 16 June 2022, from https://www.androcom.com/2020/06/10/the-pros-and-cons-of-avatars-as-on-screen-mentors/

Kinetic Games. (n.d.). *Kinetic Games*. Phasmophobia. Retrieved 13 June 2022, from https://kineticgames.co.uk/

Lehrerinnenfortbildung Baden-Württemberg ZSL. (n.d.). *Checkliste für die Präsentation*. Checkliste Für Die Präsentation. Retrieved 16 June 2022, from https://lehrerfortbildung-bw.de/st_digital/medienkompetenz/digipraes/projekt/bausteine/bau/tipps.htm

les. (2019, December 11). *Auf Plattformen: Pädophile machen Jagd auf junge Gamer—20 Minuten* [News]. Pädophile Machen Jagd Auf Junge Gamer. https://www.20min.ch/story/gamer-leiden-unter-sexueller-belaestigung-162929450820

Microsoft. (n.d.). *Microsoft Teams*. Videokonferenzen, Besprechungen, Anrufe | Microsoft Teams. Retrieved 13 June 2022, from https://www.microsoft.com/de-ch/microsoft-teams/group-chat-software

psuwara. (2018, February 6). *AssetBundles—Hype and let down, or real benefit? - Unity Forum* [Forum]. AssetBundles - Hype and Let down, or Real Benefit? https://forum.unity.com/threads/assetbundles-hype-and-let-down-or-real-benefit.516025/

PZ Bremen/Hamburg. (n.d.). *Pyrotechnische Fachkunde DMYV*. Pyrotechnische Fachkunde DMYV - PZ Bremen/Hamburg. Retrieved 16 June 2022, from https://www.pa-bremen.de/scheine/pyro/pyrotechnische-fachkunde-dmyv.html?file=files/pa-bremen/#/

SAICSIT 2001, Renaud, Karen, Kotzé, P., Barnard, A., & South African Institute for Computer Scientists and Information Technologists (Eds.). (2001). *Hardware, software and peopleware: South African Institute of Computer Scientists and Information Technologists Annual Conference, 25-28 September 2001 Pretoria, South Africa: Vol. The effects of avatars on co-presence in a collaborative virtual environment*. Unisa Press.

Subspace Team. (2021, December 14). *Why In-Game Voice Chat is Essential to Multiplayer Gaming*. Why In-Game Voice Chat Became Essential to Multiplayer Gaming and How It Impacts the Metaverse. https://subspace.com/resources/in-game-voice-chat-and-the-metaverse

TeamSpeak Systems. (n.d.). *TeamSpeak*. Home | TeamSpeak. Retrieved 13 June 2022, from https://www.teamspeak.com/en/

Ubisoft Entertainment. (n.d.). *Tom Clancy's Rainbow Six Siege*. Tom Clancy's Rainbow Six Siege | Ubisoft (DE). Retrieved 16 June 2022, from https://www.ubisoft.com/de-de/game/rainbow-six/siege

Unity Technologies. (2021, February 14). *Introduction to Asset Bundles—Unity Learn* [Tutorial]. Introduction to Asset Bundles. https://learn.unity.com/tutorial/introduction-to-asset-bundles#

Valve Corporation. (n.d.-a). *Counter-Strike: Global Offensive*. Retrieved 13 June 2022, from https://blog.counter-strike.net/

Valve Corporation. (n.d.-b). *Steam-Overlay*. Steam-Support :: Steam-Overlay. Retrieved 13 June 2022, from https://help.steampowered.com/de/faqs/view/3978-072C-18DF-FBF9

Zoom. (n.d.). *Zoom*. Videokonferenzen, Cloud-Telefonie, Webinare, Chat, Virtuelle Veranstaltungen | Zoom. Retrieved 13 June 2022, from https://zoom.us/

9. Glossary

Term or Acronym	**Description**
VR player	The user, who joins the VRLabs with a VR headset and who is usually the host of a VRLabs scene/room/game. Equivalent to the terms “VR-Host”, “host” or “primary user”. Sometimes referred to as “teacher”.
AR viewer	The user, who joins the VRLabs with a mobile device and who is usually a viewer of the VR-Player. Equivalent to the terms “AR-user”, “viewer”, “secondary user” or “observer”. Sometimes referred to as “student/students”.
Virtual Physics Laboratory	The previously made project named “VRLab”. Often referred to as “VRLabs”, not to be confused with the VR-Mock application or the VRLabs game scenes.
AR Extension	The main objective of this project, the developed AR mobile application. Often referred to as “AR-Viewer”, but not to be confused with the AR viewer.
VR-Mock	The mock application of the original VRLab. Often referred to as “VR-Player”, but not to be confused with the VR player.
PUN	Photon Unity Networking of Photon Engine
RPC	Remote Procedure Call, used in the context of PUN RPC events.
AR	Augmented Reality
VR	Virtual Reality
Compass-Feature	The compass feature is a UI element which imitates a compass. This feature enables rotation of the VR space. Sometimes referred to as “Rotation-Feature”.
Overlay-Feature	The overlay feature is a semi-transparent UI overlay with its own page-based navigation. It is used to structure and display all the detailed information about the project, such as physical or electrical configurations and calculations. Often referred to as “overlay”.
Ping-Tap-Feature	The ping tap feature is used to place a pin on a game object and share it with all other users to place a question or better describe which object is meant. Sometimes referred to as “Ping-Tap”, “Ping” or “Pin”-feature, might also be referred to as “Tap-Ping”-feature.
Solar Panel Configuration	The configuration of a solar panel group, in our case defined by rotation. Sometimes shortened to “configuration”.

10. Declaration of honesty

«We the undersigned declare that all material presented in this IP5 project report is our own work and written independently only using the indicated sources. The passages taken verbatim or in content from the listed sources are marked as a quotation or paraphrase. We declare that all statements and information contained herein are true, correct and accurate to the best of our knowledge and belief. This paper or part of it has not been published to date. It has thus not been made available to other interested parties or examination boards.»

Windisch, the 17th of June, 2022

Thierry Odermatt  …………………………….

Andreas Leu  …………………………….

Appendix

Integration Guide

This integration guide will help you to integrate our interactions into your own application.

This project was implemented and tested using Unity Version 2020.3.30f1.

Step-by-Step Integration

The following part will consist of a simple guide on how to integrate each part of the application. There will be minimal to no explanation. For detailed explanations, please refer to our project report.
If any settings of Components are unclear, please refer to our scenes.

AR

The AR part is split into two scenes, the Launcher and VRLabs.

AR Launcher Checklist

The AR Launcher can, for the most part, be left as is, but if you want to make changes, the following should be present:

- Device Manager, “IsHome” set to true
- Event System
- ControlPanel (as child of a canvas)
- ProgressLabel (as child of a canvas)
- Launcher
    - Set ControlPanel reference (instance)
    - Set ProgressLabel reference (instance)
    - Set RoomButton reference (prefab)
- MainCamera

AR VRLabs Checklist

The AR VRLabs can, for the most part, be left as is, but if you want to make changes, the following should be present:

- AR Session Origin (with AR Camera)
    - AR Tracked Image Manager (Set SerializedLibrary to AR marker image)
- AR Session
- Directional Light
- VRLabs_ARExtension
    - PhotonManager
    - EventSystem
    - Canvas
    - DeviceManager
    - VRSpace
        - Set TrackedImageManager reference (instance)
        - PlayerHeadSync (View ID to 1)
        - PointerStick (View ID to 2)
        - RightHand, “Hand” Prefab (View ID to 3)
        - LeftHand, “Hand” Prefab (View ID to 4)
    - TapManager
        - Set AR Camera Reference (instance)

Dropping in the VRLabs_ARExtension prefab necessitates you to only set the references to the TrackedImageManager and the AR Camera.

Overlay Content Integration

The content of the Overlay is currently hardcoded. To change this, go to the Script Assets/Scripts/OverlayManager.cs, line 275ff.
There you will find the fields that need their values assigned.

AR Prefabs Key

All non-default elements of the checklist are prefabs. Any objects that contain a PhotonView can be found in /Assets/Resources. Make sure all listed prefabs in /Assets/Resources are present as they will be used during runtime.

The folders are as follows:

- ARViewer/Assets/Prefabs
  - /Launcher
    - ControlPanel
    - Launcher
    - ProgressLabel
  - /SharedPrefabs
    - DeviceManager
    - EventSystem
  - /VRLabs
    - VRLabs_ARExtensions
    - Canvas
    - VRSpace
    - TapManager
- ARViewer/Assets/Resources
  - Hand (Materials in /HeadnHandResources)
  - PlayerHeadSync (Materials in /HeadnHandResources)
  - PointerStick (Materials in /PointerStickResources)
  - Mobile (Materials in /FancMobile)
  - Panel (Materials in /PanelResources)

VR

The VR part is a working demo and is meant to be integrated into an existing application. It consists of 2 Scenes. The following checklists will not include any environmental objects like floor or light sources.

We recommend finding and testing a more fitting model for the "Pin". To do this, simply replace the "PinModel" child of the Pins in both the AR and VR. Make sure to give the VR version a collider for the delete interaction.

You can replace the "Board" prefab within the different menu prefabs with a different model, if so desired. It is not recommended to change any of the names of the other objects in the prefabs, as many of them are referred to by name.

VR Launcher Checklist

It is crucial that the VRLabs scene is joined via the Launcher to correctly set up the Photon room. The Launcher scene can be any environment, as long as the following components are present:

- LauncherBoard
- EventSystem
- Player (or similar)

VR VRLabs Checklist

For the VRLabs scene the following prefabs should be present. The player can be exchanged with another that has the mentioned prefabs attached to the right places.

- AssetSelectorBoard
    - Check that the Asset Button is set to the corresponding prefab
    - Set Reference to the PanelSwitcher (VRLabsMenuBoard)
        - Necessary to delete current solar panel config on asset change
- PointerStick (View ID 2)
- Player: Any XR Rig with:
    - "Head", such that it follows the players head (View ID 1)
    - "Hand", such that it follows the players right hand (View ID 3)
    - "Hand", such that it follows the players left hand (View ID 4)
- EventSystem
- PhotonManager
- VRLabsMenuBoard
    - Set References (prefab)
    - Check that buttons call the correct Method (instantiateTypeX Button)
        - For dynamic loading of solar panels, see below

PanelSwitcher Integration

Currently the information (position, rotation) of the panel configuration is generated by having a prefab of objects that then are iterated over and their values saved to a list.

From there on it should not matter where we got the list from.
What has to be done to load a configuration:
Wherever you trigger an update to a configuration, including the initial creation, you will simply have to call the method “InstantiatePanels(...)” from the “PanelSwitcher” Script. You will need to hand it an IEnumerable with elements of type Tuple<string, Vector3, Quaternion>.
The script will then instantiate one panel for each entry.
The string must be the name of an existing and valid prefab that is located in the /Assets/Resources folder. A valid prefab means in the case of a solar panel, that it needs a PhotonView component.

AssetSelection Integration

To show a dynamically updated list of available assets, you have to replace the code in the method “GetAssetBundlesList()” in both the AssetSelector and Launcher scripts.
Currently this is simply a hardcoded list of strings.
The strings have the following significance:

- They will be displayed “as is” in the list of selectable assets
- They need to match the name of the assetbundles exactly, except case-sensitivity
    - Example of valid String / assetbundle pairs:
        - “Projekt ROT” / projekt rot
        - “turnhalle” / turnhalle
    - Example of invalid pairs:
        - “Project Red” / projekt rot
        - “turn halle” / turnhalle

You can for example have the method GetAssetBundlesList() return the list of filenames on a server or similar.

Changing the list of Room Names

If you want to change the list of verbose room names to a more desirable selection, you can simply replace or edit the Assets/StreamingAssets/RoomNames/RoomNames.csv. Make sure to have one room name per line. It is crucial that this file remains within the StreamingAssets, or else it cannot be referenced by path from the script anymore.

VR Prefabs Key

All non-default elements of the checklist are prefabs. Any objects that contain a PhotonView can be found in /Assets/Resources. Make sure all listed prefabs in /Assets/Resources are present as they will be used during runtime.

- VRPlayer/Assets/Prefabs
  - /Launcher
    - LauncherBoard
  - /SharedAssets
    - AssetButton
    - EventSystem
  - /VRLabs
    - AssetSelectionBoard
    - VRLabPhoton
    - VRLabsMenuBoard
- VRPlayer/Assets/Resources
  - Hand
  - Head
  - Mobile
  - Panel
  - Pin
  - PointerStick

Assetserver

The Assetserver is a unity project that allows you to create assetbundles that then can be loaded dynamically during runtime. The current implementation uses GitHub Pages to host them, for more details on hosting and its requirements, refer to chapter "AssetSelection Integration" in this document.

To create new assetbundles, follow these steps:

1. Open the AssetbundlesGenerator Unity Project.
2. Create **One** folder per assetbundle you want to create
   - Preferably in the folder /Assets/Prefabs/AssetBundles
   - Make sure the main asset is called "WebAsset"
   - Make sure that all materials, meshes and other referenced object are within that same folder
     - When duplicating folders, make sure to reference the new instance of the object
3. Select your folder
   - The one with the new asset
   - Make sure to select it from the Project View Content part and not the Project View Folder Hierarchy
4. In the inspector of the Folder, find the AssetBundle field at the bottom
   - It should say None by default, or the name of an asset bundle
5. Select your assetbundle name from the dropdown menu
   - Specifically the left of the two dropdown menus
   - If your assetbundle name does not yet exist, select "New…" in the dropdown
   - Asset Bundles are all lowercase letters and can contain spaces
6. Once all your assets are up to date and ready, Select Assets/Build AssetBundles from the top menubar (Unity MenuItem)
7. Wait for the Asset Bundles to finish building
   - The AssetBundles are saved into the assetbundles folder, which is at the root of the git hierarchy
   - You can select a different directory by changing the script "Assets/Editor/CreateAssetBundles".
     - Change the variable _abDirectory to your desired directory
     - The relative path starts at the Unity root (the one with the "Assets" folder)
   - Assetbundles are files without file endings
   - The former Asset Bundles get renamed to <filename>_OLD, in case you need to restore them
8. Upload your assetbundles
   - With GitHub Pages this means simply committing and pushing the changes

Lookup Tables

The following tables are meant as a quick reference for crucial information, such as IDs and Important Strings.

Photon View ID

These are the manually set Photon View IDs. Changing these would require you to change them in AR and VR and would require all users to update their application. Thus, this is a breaking change.

Prefab/Object Name	View ID
PlayerHeadSync	1
PointerStick	2
RightHand	3
LeftHand	4

User Testing Conduction Template

Four groups of testers, starting at 2 pm, 3 pm, 4 pm and 5 pm on the 8th of June.
We use two modes of testing, AR test and lecture simulation. If there is enough time, the testers may take turns in playing the VR host.

S1: AR test:
In the AR test mode, the moderator will play the VR user, and the tester will focus on the AR application. This mode will be simulating a lecture, where the teacher is leading and the students can observe and interact in AR to solve the tasks given by the teacher.

S2: Lecture simulation:
In the lecture simulation mode, the testers will play the VR application as well as the AR application. This mode will be simulating a lecture, where students have to work together to solve tasks given by the moderator.

Group A, S1:
- 2 pm
- __ Participants

Group B, S2:
- 3 pm
- __ Participants

Group C, S1:
- 4 pm
- __ Participants

Group D, S2:
- 5 pm
- __ Participant

How to test/animate testers, what tasks…

<u>Introduction:</u>
In the last two weeks, you have learned about solar power's physical and electrical characteristics and different solar panels. As of now, your teacher gives you the opportunity to play around with different configurations of solar panels and the buildings they are seated on, to give you a better understanding of how solar power works and what influences the effectiveness of solar panels.
[S1]
Your teacher will be playing with the VR headset, you can follow with the AR application on your mobile device.
[S2]
Your teacher will observe the workshop, one of your group may play the VR user, and all others can follow on the mobile device. You can take turns playing the VR application.
(To install the application, download the .apk from this URL: *https://vrlabs-ip5.github.io/assetbundles/builds/demo.apk*)

<u>Main part:</u>
The VR player and the AR user have several tasks to absolve.

VR:
- Open a game room
- Have a look around and get familiar with the game world and observe if all other students join the room.
- You have now talked about the building, time to load the first solar panel configuration.

- Let’s say the VR player wants to talk about a specific solar panel, somewhere in the middle. Find a way, so that all users of the group understand which panel is meant.
- Let’s say you want to make a simple poll, for example: “Should we load the next panel configuration?”. Ask the students to place them either left or right of you, so that two groups are built, one for “Yes”, and one for “No”. Does it work? How easily does it work?
- Load the next panel configuration
- Load another project/building

AR:
- Join the game room the VR player has opened
- Scan the marker to position the VRSpace correctly
- Have a look around and get familiar with the game world and observe if all other students join the room.
- Let’s say an AR user has a question or a note about a specific solar panel. Find a way, so that all users know which panel is meant.
- As an AR user, you have the ability to see and observe all configurations and calculations of the world/project, find the following information, in detail, find:
    - Room Name (Raum)
    - Project Name (Projekt Name)
    - Energy in DC, AC (DC Energie, AC Energie)
    - Power (AC Leistung)
    - Panel Type
    - Energy production in a year (Energieproduktion in einem Jahr)

<u>Cool down:</u>
- Fill out questionnaire
- Open discussion with the participants to receive valuable open feedback, offering a drink and some snacks as a thank you.
- talk about:
    - overall experience

      VR:
        - other AR users
        - configuration boards
        - pointing stick
        - ping interaction of AR users

      AR:
        - launcher
        - VR user
        - other AR users
        - rotation
        - scaling
        - VR pointing stick
        - overlay (more detailed information)

User Testing Questionnaire Template

Demographics

1. **Date:** ________________
2. **First Name:** ________________ **Last Name:** ________________
3. **Gender**: O f O m O other
4. **Age**: O 0-19 O 20-29 O 30-49 O 50+
5. **Prior experience with VR:**
   (virtual reality)
   O Layman (0 hours)
   O Newcomer (1-10 hours)
   O Enthusiast (11-100 hours)
   O Expert (100+ hours)
6. **Prior experience with AR:**
   (augmented reality)
   O Layman (0 hours)
   O Newcomer (1-10 hours)
   O Enthusiast (11-100 hours)
   O Expert (100+ hours)

Overall Feedback Questions

7. **This application was fun to use.**
   strongly disagree O —— O —— O —— O —— O strongly agree

8. **It was easy to interact with the other users within the app.**
   strongly disagree O —— O —— O —— O —— O strongly agree

9. **I think that I would like to use this app in an educational context.**
   strongly disagree O —— O —— O —— O —— O strongly agree

10. **I think that the app is unnecessarily complex.**
    strongly disagree O —— O —— O —— O —— O strongly agree

11. **I learned to use this app very quickly.**
    strongly disagree O —— O —— O —— O —— O strongly agree

12. **I found that the features in the app were well integrated.**
    strongly disagree O —— O —— O —— O —— O strongly agree

13. **I found that the features were easy to use.**
    strongly disagree O —— O —— O —— O —— O strongly agree

14. **I would need a tech supporter to use this application.**
    strongly disagree O —— O —— O —— O —— O strongly agree

15. **Within seconds, I knew how to rotate the game world.**
    strongly disagree O —— O —— O —— O —— O strongly agree

16. **Within seconds, I knew how to scale the game world.**
    strongly disagree O —— O —— O —— O —— O strongly agree

17. **Within seconds, I knew where to find additional settings and configurations.**
    strongly disagree O —— O —— O —— O —— O strongly agree

Open Questions, Open Feedback

18. **What did you most like about this application?**

19. **Which feature(s) did you most like?**

20. **What feature(s) did you miss?**

21. **What was the biggest pain point?**

22. **Is there anything else you would like to tell us?**

Thank you for participating in our User Testing Workshop!

Please return this questionnaire to your moderator.

User Testing Disclaimer Template

Thank you for participating in our usability test of the project “VR-labs on smart devices for education“. By your participation, you help to further develop and optimise our application and allow us to better meet the needs of our customers and end-users.

The feedback you give us during the workshop is very valuable for us, especially if it is constructive and critical. Please feel free to share any thoughts, positive or negative, with us.

For evaluation purposes, we will record the test session (audio and video). Any processing of these recordings will be done anonymously.
O I agree that the audio and video recordings will be stored for later evaluation.
O I also agree that excerpts or transcripts of these recordings may be used for internal purposes (e.g. for presentations).

You have the right to withdraw your consent to the above selection at any time. To do so, please inform Thierry Odermatt (thierry.odermatt@students.fhnw.ch) or Andreas Leu (andreas.leu@students.fhnw.ch).

User Testing Findings

AR - Launcher:

- fast and easily understandable
- Scrollbar is visible if list is empty/short, empty list is scrollable
- InputField does not support UTF-8 encoding, some characters cannot be typed in

AR - VRLabs:

First impression:

- For first users of AR, a tutorial to scan the marker would be needed
- For first users of AR, without a tutorial/introduction it is difficult to
- Without any first-time tutorial/introduction screen, it is not given that the user understands, that the application runs in landscape mode
- Without any first-time tutorial/introduction screen, the users can only know about interactions by trying out and playing around

Basic Interactions (Rotation/Scale):

- Rotation/Scale feature found and used easily
- Users were quickly able to reposition themselves in the game world to find the VR user

Compass-Feature (Rotation in detail):

- For most users, the first impression of the compass feature was unclear. After trying it out and playing around, all users quickly understood what the feature does and could identify that it should resemble a compass.
- to help identify the compass, celestial direction marks would be needed
- Idea: instead of a big compass, display a small whole compass at the top right

Find hidden VR user:

- users could easily reposition and find the VR user

Describe, which panel the VR user meant (hands/pointing stick):

- the users could quickly determine which panel the VR user told, explained by hands as well as pointing stick

Describe to all other users, which panel the AR user means (tap-ping):

- Ping mode could not be found without a hint
- Ping mode is described as difficult to use without any tutorial/introduction
- Ping mode was often confused with “Location”, as the icon used for ping mode was a location icon
- ping mode does not work as expected, difficult to understand what ping mode should do without a tutorial/introduction
- long press is counter-intuitive
- hold to instantiate a pin is counter-intuitive
- The colour and object used for a ping are not easily visible and the size after sent is not optimal
- It could be easier to understand for the users if a preview ping is placed in the middle of the screen when ping mode is active

- Maybe with several users setting pings, it would make sense to have a label above the pin which shows the creator of the label
- (Bug) send ping would delete the instantiated ping but show it for other users

Find detailed information about the project (overlay):
- Overlay was found quickly and could be used fast and easily
- Naming of the “Einstellungen” page feels misleading, better maybe “Allgemeines”
- to close the overlay, for mobile users it would be intuitive to click beside the open overlay
- Discussion between testers: the current navigation in the overlay is based on two buttons, forward and back at the top of each page. This is closer to games than mobile applications. In mobile applications, what one would expect is a drawer menu with each chapter as a title to click on, load the selected page with the drawer menu at the top left. This would help to give the user an overview and quick access to all chapters available. Which implementation would fit better?

Other AR users:
- (Bug) The front and back nickname text of other AR users are both shown at the same time (faulty shader)

Overall impression:
- little to no interaction between AR users, little interaction in general
- devices will get quite hot

Further interaction/tool ideas:
- Measurement tool (user can set two points to display the length between)

VR - Launcher:
- fast and easily understandable
- (Bug) User can teleport behind the launcher wall

VR - VRLabs:
- Found very “cool” and helpful to see the AR users as mobile phones
- Difficult to pick up the pointing stick
- (Bug) Ping feature does not behave as expected, pins get deleted if other users send new pings
- (Bug) The pointing stick can move pin

Open Questions/Open Feedback of Questionnaire

What did you most like about this application?
- One VR user, all the rest with mobile AR devices
- easy and straightforward to use
- pointing stick

Which feature(s) did you most like?
- Pointing stick
- Ping-Feature
- That VR and AR users can be in the same game room

What feature(s) did you miss?

- Configurations should also be visible as I long press a solar panel
- user list of all users within the room
- Minimap: a small map where I can see my location in relation to the VR user, other users and the building
- Manual rotation of the model by finger gestures

What was the biggest pain point?

- to grab the pointing stick in VR
- menu navigation with arrows
- ping feature: misleading icon; need to press to make pin appear
- ping feature: unintuitive
- ping feature: difficult to set the ping precisely

Is there anything else you would like to tell us?

- Overlay: why is it named settings if there are no settings, you get room infos
- Overlay: menu navigation could be better, arrow navigation is counter-intuitive

Analysis of Questionnaire		**Edi**	**Katrin**	**Natascha**	**Lucas**	**Total**	**Mean**	**Max**	**Min**	**Midd**
Demographics										
Gender (m/f/o)		m	f	f	m					
Age		20-29	30-49	20-29	20-29					
VR Experience		100+	0	1-10	1-10					
AR Experience		11-100	1-10	0	11-100					
Overall Feedback Questions										
This application was fun to use	+	3	4	3	4	14	3,5	20	4	3
It was easy to interact with the other users within the app	+	5	2	2	4	13	3,25	20	4	3
I think that I would like to use this app in an educational context	+	5	5	5	5	20	5	20	4	3
I think that the app is unnecessarily complex	-	1	2	1	2	6	1,5	20	4	3
I learned to use this app very quickly	+	5	4	5	5	19	4,75	20	4	3
I found that the features in the app were well integrated	+	5	4	4	3	16	4	20	4	3
I found that the features were easy to use	+	4	4	4	4	16	4	20	4	3
I would need a tech supporter to use this application.	-	1	3	3	1	8	2	20	4	3
Within seconds, I knew how to rotate the game world.	+	5	4	5	4	18	4,5	20	4	3
Within seconds, I knew how to scale the game world.	+	5	5	5	4	19	4,75	20	4	3
Within seconds, I knew where to find additional settings and configurations	+	5	1	5	4	15	3,75	20	4	3
9 positive qestions, 2 negative questions, 5 points per question, best score results in 47, worst score results in 19		44	38	42	40		41	47	19	33

Windisch, 18.05.22

Information on the project procedure & project agreement IP5, 22FS_IIT04 VR-labs on smart devices for education

Supervisor: Arzu Cöltekin
Cloe Hüsser

Client: Renato Minamisawa

Project duration: 22.02.2022 until 17.06.2022

Task

1. Familiarization

1.1 Expectations for the project process

Dates
Fix appointments early, i.e. reviews with the customer and about every 2-3 weeks a meeting appointment with your supervisors. Clarify any absences right at the start of the project.

Meetings
Meetings are generally intended to discuss the current status of the project, clarify questions, discuss ideas and plan the next steps.
Send a list of agenda items and all other necessary documents to the supervisors. At the beginning of each project meeting, explain the current status of the project, the progress and problems as well as the planned steps.
You can use the meetings by arrangement and, if necessary, also for specific questions (e.B micro-teaching, brainstorming, presentation of results or mentoring). However, come to a meeting with as specific questions as possible.
Please record the discussed contents and decisions in a timely manner.

1.2 Specifications for the agreement

As a first task in your work you have to complete this agreement (cf. point 3). A first version should be produced by 2-4 weeks (BB 4-6 weeks) after kick-off. For projects that require technical analysis, it may be useful to carry out a first implementation iteration before the sub-mission of the project agreement. Please complete the following items:

Initial situation
Formulate the project and the initial situation in your own words.

Project vision
Describe which goals and results are to be achieved with the project. The vision serves to derive quality criteria.

Project specific issues
In addition to the general questions, formulate 2-3 project-specific questions. These serve as a basis for a scientifically structured research and the derivation of suitable solutions.

Examples of questions and solutions:

- Which approaches do you use to reach the defined target group?
  Solution approach: Development of concepts for user-centered approaches and implementation of the user interface of the application, e.g. in the form of storyboards with a continuous user story or GUI prototypes.
- With which technical concept do you achieve the desired solution?
  Solution approach: Technology evaluation, development of technical solution concept (PoC), definition of subsystem decomposition, architectural style and technologies.
- Which interaction concepts, interface designs and visual languages are suitable for your approach?
  Solution approach: Development of interaction concepts and graphically carefully designed, clearly structured imagery for interface design, which meet the requirements of an innovative user experience.
- With which technical implementation do you meet the requirements for functionality, usability, reliability, efficiency and maintainability?
  Solution approach: Implementation of a executable application for a previously evaluated setup and defined usage scenario based on suitable technologies and frameworks
- Correctness, usability and reliability are central to the successful introduction of the software. How can you ensure and test them?
  Solution approach: In-depth testing of correctness, usability and reliability, documentation of Test results, demonstration of the fulfillment of the requirements by means of live test.

Methodology
Describe how the goals are achieved. Which methodologies do you use for this (e.B. Scrum, Agile, scientific approach, etc.).

Planning
Create an initial project schedule. Define work packages and their deliverables.

Risk Assessment
Identify and evaluate risks within the project and develop strategies for dealing with them.

2. Documentation

2.1 Written documentation (Thesis Rapport)

Document in writing and electronically your approach, the theoretical background, the application of methods and concepts, the implementations and test results. Also check the planned with the actual schedule, the achievement of goals and reflect on experiences.
Be sure to strictly separate personal comments from facts. **The main part of the documentation is completely fact-based**. This means that no sentences of the kind "Then we had the problem x and tried to solve it with y" are allowed to occur. But if such a problem x really exists and not only you did not get to the edge with it, then you should write: "Tests z have clearly shown that a problem x exists. Possible approaches to solve problem x are a, b and c. We chose variant c for reasons e and f." Only in an extra section can you formulate your personal impressions, experiences, problems and the like.
It is also important that a good documentation must still be read after many years and that it gives the reader a well-rounded picture, even if he was not directly involved in the work. Please also attach great importance to linguistic quality.
The target audience of this documentation are the supervisors, the experts, the client and future students who want to continue working in this area.
The documentation is created during the course of the project. For the second coaching meeting, a table of contents of the report should be prepared so that it can be discussed with the supervisors.
The parts for research and analysis are to be presented after the first third of the project.

On the web portal of the FHNW you create a project presentation (web summary). For bachelor theses in the spring semester, you will also create a poster for the exhibition. Both artifacts must be discussed with the supervisors prior to publication.

The following information must be mentioned on all publications:
- Logo FHNW
- Semester project IP5 or Bachelor thesis (IP6)
- Project name
- Spring- or Autumn Semester 202x, Degree Programme Computer Science (Profiling iCompetence), University of Applied Sciences and Arts Northwestern Switzerland
- Submitted by: Name of Students
- Submitted to: Name of Supervisor
- Client: Company / Institution
- Date

Further information on writing reports can also be found on the Information Literacy Platform

2.2 Presentations

Presentations take place in consultation with the supervisors and the client. The expert will also be present when defending your bachelor's thesis.

On the one hand, presentations provide an overview of the entire project and the results achieved and deepen one or two important interesting questions. Also part of the presentation is a concise demonstration of how to use your software. With the audience, you can expect a technically experienced professional audience. Schedule 30' for the presentation and demonstration and reserve 30' for questions and discussion.

2.3 Publication of the project results

If the work or parts of the work are published, all names of the project participants (students, supervisors, clients) as well as the name of the institution (FHNW) must be mentioned. Before each publication, supervisors and clients must be asked for their consent in advance.

2.4 Protocols

Protocols are an important part of the documentation. Professionally managed protocols contain the following points:
- Date, Space, Time, Participants, Excused
- Agenda
- Project status (possibly with screenshots, sketches, etc.; Status according to planning)
- Content (fact-based, thematically structured and comprehensible in terms of content; Decisions are recorded)
- Open questions
- Next steps; Appointments & tasks (who, what & until when)

2.5 Document repository

Set up access to your document storage for the maintainers. If there are no compelling reasons against it, use the Gitlab infrastructure of the FHNW[1].
Also, use this document cabinet to store additional documentation, such as how to run your code. Make sure that an adequate commit history is visible to the caregivers.

2.6 Submission

[1] https://gitlab.fhnw.ch/

The project submission includes (unless otherwise defined with the project manager) the following artifacts:

- Written documentation (Thesis Rapport)
- Project agreement (on the shelf as an appendix in the thesis)
- Codebase (documented & with readme to explain the setup),, hosted on GitLab of the FHNW (https://gitlab.fhnw.ch/iit-projektschiene/[semester]/[project]) and as a ZIP archive
- Link to the project appearance on the FHNW web portal
- other artifacts, if available (screencast recommended, ...)

3. Project-specific agreement

3.1 Starting situation

VRLabs is a VR application that allows to observe and modify a setup of solar power cells placed on a virtual building. The setup is simulated on a server hosted by Solextron. As current hardware serves the Oculus, due to it being an affordable and portable solution.
The Project was originally created on Godot but is currently being ported to the Unity Engine. The porting is not within the scope of this project. VRLabs uses OpenXR as the XR Interface for VR. Currently, the building and solar panel placement is done on the Solextron Webpage. The setup is then saved as a JSON file, which can be transmitted to the VR environment to be viewed. Interactions in VR include only changing the panel type so far. Tilting, moving, adding, or removing panels as well as modifying the building are not yet supported interactions.

The current use-case of the application is within education. The teacher presents the experiments by using the VR headset and displaying them on a bigger display to view or hands out headsets to have the students test the experiments by themselves. The problem is that VR headsets are expensive and therefore a solution has to be developed to get an inexpensive way to allow students to view and participate in the experiments without needing a headset. Based on this, the primary VR headset based user has to be extended by mobile device viewers and possible interaction concepts with the primary user have to be developed.

3.2 Project vision

Our overarching vision for the system, with disregard to feasibility, contains the following features:

1) A primary user in VR
   - VR headset (likely Oculus Rift S due to portability)
   - Full interaction capabilities
2) Many secondary users in AR
   - Phone or by using web-based AR
   - Lightweight interaction model
3) Topologically correct and visually pleasing surroundings

As the first part is already done and extending the VR interactions is not part of this project, we only must tailor the rest of the system to fit with the VR application.

We assume the second part to be within the possibilities and expectations of the project. In a first step, the primary user will be extended with secondary users using mobile AR or web-based AR. Therefore, the virtual world of the primary user and its interaction must be shared with the secondary users. To achieve this, we hope to rely on Unity collaboration asset packs, which will handle a lot of the synchronization and networking problems. In this first phase, the secondary users will not have any further interactions. To allow the user to create their own building eg. by the web application, the building must be hosted externally on a server/cloud. Therefore, we will develop this functionality and integrate the current calculation server to display information about the experiment.

In the second phase, we elaborate on possible interaction models for secondary mobile users and will use a user-centred design to determine 1-2 fitting interactions to be implemented.

The third part, while included in the broader conceptual vision, is not in the scope of this project, and would require another project of a similar scale, thus can be considered as future work.

3.3 Questions

A. How can mobile devices with AR support a VR-based application?

B. What interactions are possible between users of the mobile devices and the user of the VR system?

C. How can Unity Assets (3D models) be modified, exchanged and synchronized at runtime?

In addition to the project-specific questions, the following generic questions will be considered in the implementation of their work:

D. Identification of suitable scenarios and user interface prototyping: Which approaches do we use to reach the defined target group of Teacher-Students (1-m)?

E. With which technical concept can we achieve a synchronized solution for AR and VR users?

F. Which interaction concepts, interface designs and visual languages are suitable to give the AR-Viewers enough options and interactions to follow the guide of the VR user?

G. With which technical implementation do we meet the requirements for functionality, usability, reliability, efficiency and maintainability?

H. How can we ensure and test the correctness, usability and reliability of our implementation?

3.4 Methodology

In this project, work is to be done in an agile manner. The fast iterations will create timely solutions and enable quick correction of errors, and regular contact and meetings with the customer and the supervisors will ensure the project's direction and development.
The planning, as well as the upcoming tasks, are to be created and sorted via a Kanban board so that a quick and constant overview of the project status can be obtained. With the help of this task-based or issue-based project overview, the software development, as well as the tasks in all other areas such as requirements engineering or scientific work, are ensured and the user-centred design is ensured through regular testing of the theoretical and practical prototypes with the target group. User tests will be in a planned matter with a given scenario with ca. 5 possible end-users at a time. For the technical side, we will use Unity 2020.3.30f1 as a base together with Photon Engine (Photon Unity Networking - PUN 2) to make multiplayer possible. With PUN 2, we are able to use a strong and widely used framework for unity networking with all its benefits like standardized network traffic optimization. Together with unity comes a collection of frameworks, tools and packages which will help us in a variety of ways: Besides the preinstalled packages, for AR we will use Unity's AR Foundation together with ARCore XR Plugin, to verify the quality of the selected AR marker, we will use the "arcoreimg" tool from ARCore, for the user interface we will use Unity's Unity UI as well as Unity's TextMeshPro to create more pleasing ui and for android debugging and logging we will use Unity's Android Logcat integration. As Unity works on C#, the main programming language will be C# for scripting. For usability, usability testing and user acceptance, our validation approach will be by moderated usability tests in a controlled environment, as the use case of our application will be within a classroom, which means that there will most likely be a somewhat controlled environment.

3.5 Planning

The most important and largest work packages (effort estimated in person-hours):

MS1: Minimum viable product (MVP)	
Goals	The VRLab will be expanded to include a spectator mode based on AR.
Activities	WG1: Set up a demo project for VR to simulate the existing VRLab. WG2: Set up demo project for AR to learn AR programming on Unity. WG3: Research on viewer mode WG3: Implement viewer mode

	WG3: Test viewer mode
Deliverables	AR-based viewer mode
Start	CW 9
End	CW 13
Effort	Approx.. 80-90 h

MS2: Interaction concepts/interaction models	
Goals	For VRLabs, different interaction concepts/models for the interaction between presenter and viewer are developed
Activities	WG4: Brainstorm and collect concepts WG4: Deal with concepts and choose the best two ("best" in the sense of: realizable, useful, added value for users) WG6: Theoretically elaborate concept 1 and put it into practice. WG5: Theoretically elaborate concept 2 Test concepts 1 and 2
Deliverables	Min. one concept implemented and ready for migration to VRLabs Min. one concept worked out theoretically in detail Further concepts described and theoretically worked out
Start	CW 13
End	CW 21
Effort	Approx. 220-230 h

MS3: Project end/project delivery	
Goals	Finalization and submission of theoretical and practical work
Activities	WG7: if VRLabs has already been migrated to Unity: Adding the implemented concept to VRLabs Bug fixing of the implementation Finalizing the writing (proofreading, formatting, etc.)
Deliverables	Final theoretical work Final practical work
Start	CW 21
End	CW 24
Effort	Approx. 40-50 h

WG1: Set up a demo project for VR	
Goals	It is possible to work on Unity in VR and build scenes
Activities	Set up VR demo scene Create and ensure git repo and collaboration
Deliverables	Demo scene in VR, built for the Oculus Rift S
Start	CW 9
End	CW 10
Effort	Approx. 10 h

WG2: Set up a demo project for AR	
Goals	You can work on Unity in AR and build scenes.
Activities	Set up an AR demo scene Learning to program on AR
Deliverables	Demo scene in AR, built for Android devices
Start	CW 9
End	CW 10
Effort	Approx. 15 h

WG3: Develop viewer mode on AR for VR	
Goals	A VR based scene can be extended through AR to achieve a presenter-spectator model.
Activities	Research and theorize how a spectator mode can be enabled in AR to VR. Implement the spectator mode in AR Initial testing (internal and external)
Deliverables	Viewer mode implemented and testable
Start	CW 9
End	CW 10
Effort	Approx. 55-65 h

WG4: Find interaction concepts	
Goals	Find as many possible interaction concepts for presenter-spectator interaction as possible.
Activities	Brainstorm to find as many concepts as possible Document and briefly describe each concept Work out the feasibility of each concept in terms of practical implementation Document the advantages and disadvantages of each concept Select concept 1: Most advantageous and practically feasible Select concept 2: Most advantages and theoretically testable
Deliverables	A compilation of possible interaction concepts
Start	CW 14
End	CW 15
Effort	Approx. 30 h

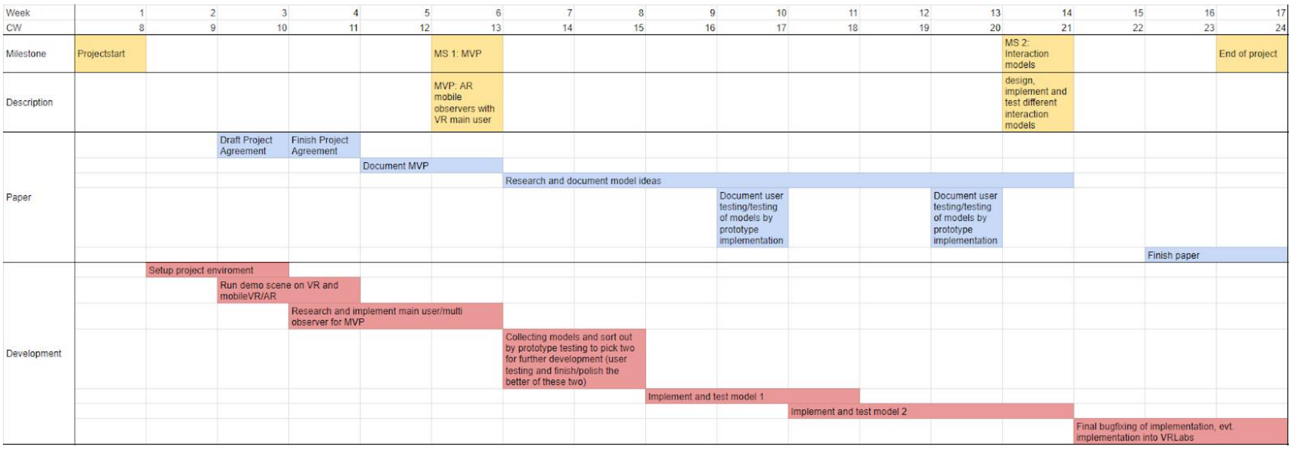
WG5: Theoretically elaborate interaction concept 2	
Goals	Interaction concept 2 dealt with in detail theoretically and tested using prototypes
Activities	Research and theoretical elaboration of the concept Prototyping the concept

	Testing the prototype of the concept
Deliverables	Theoretical elaboration of the concept Evaluation of user testing by prototype
Start	CW 16
End	CW 18
Effort	Approx. 80 h

WG6: Theoretically elaborate and practically implement interaction concept 1	
Goals	Interaction concept 1 dealt with in detail theoretically, practically implemented and tested
Activities	Research and theoretical elaboration of the concept Practical implementation of the concept Testing the implementation (user testing with ca. 5 possible end users and evaluation and documentation of the user test)
Deliverables	Theoretical elaboration of the concept Practical implementation of the concept Evaluation of the user-testing of the implementation
Start	CW 18
End	CW 21
Effort	Approx. 100 h

WG7: Integrate interaction concept 1 into existing VRLabs	
Goals	Integration of the integration concept in VRLabs
Condition	VRLabs has already been probed on Unity
Activities	Final bug fixing of the implementation Integration into VRLabs
Deliverables	VRLabs extended by interaction
Start	CW 22
End	CW 24
Effort	Approx. 25 h

Detailed project planning:
https://docs.google.com/spreadsheets/d/1yU_zk4jvY3JvBo8JOGcPuCY1grZAm3gtd2P3JrMFyDM/edit?usp=sharing



More detailed task planning:
https://github.com/orgs/VRLabs-IP5/projects/1
(For access approval problems, please contact andreas.leu@hotmail.com)

3.6 Risk Assessment

Risk	**Measure**	**Priority**	**Probability**
Feasibility of the MVP: The MVP is much more difficult to implement than assumed at the beginning and needs much more time.	Prompt research on the possible implementation to be able to better estimate the time required.	High	High
Know-How Deficits among developers due to lack of familiarity with VR/AR applications and their interaction.	Timely research and completion of tutorial courses to build up basic knowledge as quickly as possible	High	Medium
Failure to meet milestones due to fault in planning or delay	Regularly review and adjust the project planning to be able to react quickly to it	Low	Medium
Consideration of incorrect requirements due to misunderstandings	Regular consultation and updates with client to quickly clarify any misunderstandings	Low	Low
Loss of data	Data backup for documents via Google Drive and for code via GitHub	Low	Low
High latency in data synchronization between VR and (mobile) AR devices	Clarify early on what latencies are to be expected for which network framework and act accordingly	Medium	Medium
Div. problems due to miscommunication	Regular meetings and close informal communication and protocols to ensure good communication	High	Low

4. **Final provisions**
The undersigned acknowledge that they have read and understood the text and undertake to comply with the points listed and the general duty of care with their signature.

Windisch, the ...18 May 2022................

Supervisors

Arzu Cöltekin ....

Cloe Hüsser ....

Students

Thierry Odermatt ....

Andreas Leu ....

Fachhochschule Nordwestschweiz

Design UX | CG IP

22FS_IIT04: VR-labs on smart devices for education

			Priority 1	Priority 2
Advisor:	Arzu Cöltekin Cloe Hüsser	**Work scope:**	P6 (360h pro Student)	P5 (180h pro Student)
		Team size:	1er oder 2er Team	2er Team
Languages:	German or English			
Study course:	Computer Science			

Initial position

This project seeks to expand an existing 'virtual physics laboratory' that enables students to conduct experiments collaboratively that would be otherwise very difficult, expensive, or simply impossible in the real world. Furthermore, the use of virtual reality (VR) and immersive systems in the context of learning is shown to provide memorable and engaging experiences, which has pedagogical value. Last but not least, developing interactions between different participants that allow all students of a class to work together in a VR setting specifically for educational purposes brings the latest human-computer interaction knowledge into a meaningful new application at the service of students at the FHNW.

Background vector created by rawpixel.com & VRLab Screenshot

Objective

The objective of this project is to extend the current virtual lab that runs on VR headsets into a smartphone (primarily Android) application where augmented reality (AR) and 'cardboard VR' prototypes are created. Being able to work with extended reality features in their smartphones removes the dependency to headsets, and ideally allows people to interact with each other and conduct physical experiments even if they are in physically separate locations. For this to function, the users should be able to interact with each other either in person, or a VR environment (e.g., as avatars, or through integrated video calls).

Problem statement

- How can collaborative AR and VR application for smartphones (primarily Android) be designed to enable interaction with other users?
- How should the actions of the smartphone VR/AR users be visualized for other users? What are interaction and visualization designs that work well?
- How can a smartphone AR/VR application support an existing VR environment where the users can conduct physical experiments in a classroom?
- Do smartphone VR/AR applications provide additional value over the existing headset-based VRLab?

Technologies/Technical emphasis/References

- Unity 3D, Godot, CityEngine or similar
- Human-computer interaction paradigms
- Mobile VR and AR (primarily Android)
- User experiments

Note

This project is associated with a larger 'Lehrfonds' project where we will collaborate with the Physics department.