TU WIEN Informatics

# Design and Development of an Immersive Collaborative Geographical Environment for Tactical Decision-Making

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieurin**

im Rahmen des Studiums

**Visual Computing**

eingereicht von

**Bettina Schlager, BSc.**
Matrikelnummer 01327281

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Mag.rer.nat. Dr.techn. Hannes Kaufmann
Mitwirkung: Univ.Lektor Dipl.-Ing. Dr.techn. Anton Fuhrmann

Wien, 10. Oktober 2021

| | |
|---|---|
| Bettina Schlager | Hannes Kaufmann |

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Informatics

# Design and Development of an Immersive Collaborative Geographical Environment for Tactical Decision-Making

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

**Diplom-Ingenieurin**

in

**Visual Computing**

by

**Bettina Schlager, BSc.**
Registration Number 01327281

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Mag.rer.nat. Dr.techn. Hannes Kaufmann
Assistance: Univ.Lektor Dipl.-Ing. Dr.techn. Anton Fuhrmann

Vienna, 10th October, 2021

Bettina Schlager    Hannes Kaufmann

# Erklärung zur Verfassung der Arbeit

Bettina Schlager, BSc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 10. Oktober 2021

Bettina Schlager

# Acknowledgements

For starters, I would like to express my gratitude to Hannes Kaufmann, who provided advice and insightful remarks on my structure and writing. Nowadays, time is even more precious than ever, and I am thankful for Hannes, who allocated his time and expertise.

Further, I cannot begin to express my thanks to my longtime advisor and colleague Anton Fuhrmann, who guided me through countless prototypes and projects during my time at VRVis and refocused my mind whenever I got side-tracked by minor problems. The courses at TU Wien introduced me to the tools to work on VR and AR projects, but Toni showed me how to properly use them in practice.

One fact most academics can agree on is that proper academic work needs *focus, motivation and perseverance*. I want to offer my heartfelt gratitude to the people who empowered me through the process and ultimately provided the *focus* and energy: my colleagues at VRVis who spend hours on debugging, testing, proof-reading, coffee-break-support, and listening to my ideas. Marina Medeiros, Daniela Stoll and Katharina Krösl, to name a few. Further, I would like to extend my sincere thanks to VRVis and our engaging partners at the Institute for Military Geography (IMG), who enabled this project in the first place and provided the *motivation*.

At last, I am deeply grateful for my family, closest friends, study partners, and, most importantly, my loving husband (chip chap), for their endless support throughout the last years. With their help, I *persevered* through my emotional local minima.

---

This thesis was conducted at the research group Multiple Senses at VRVis GmbH in Vienna with the support of Daniela Stoll and Marina Medeiros. The primary design and implementation, such as the system and algorithmic design, 2D and 3D user interfaces and interactions and data management is implemented by the author (Bettina Schlager). Daniela Stoll supplied an initial design for the desktop user interface, and Marina Medeiros implemented parts of interactions in VR, as terrain measurements and marker placement.

VRVis, Research Centre of Virtual Reality and Visualization, is funded by BMVIT, BMDW, Styria, SFG, and Vienna Business Agency in the scope of COMET - Competence Centers for Excellent Technologies(854174), which is managed by FFG. (https://www.vrvis.at/)

# Kurzfassung


Die Planung strategischer Militäreinsätze im Zuge einer Stabsübung ist ein strukturierter und komplexer Prozess an dem interdisziplinäre ExpertInnen mit unterschiedlichen Zielsetzungen beteiligt sind. Ein entscheidender Teil des Entscheidungsprozesses bei geografischen Planungsaufgaben ist das räumliche Verständnis des Einsatzgeländes und der umliegenden Infrastruktur zur Vorbereitung von Weisungsbefugten und deren Einsatzkräften vor Ort. Analoge Planungswerkzeuge für die Stabsübung sind Geländemodelle, Sandkästen und Landkarten aus Papier und digitale Pendants sind Geografische Informationssysteme (GISs). Ein Nachteil von traditionellen Werkzeugen zur taktischen und strategische Geländeanalyse ist die fehlende, intuitive Übertragung von räumlichem Verständnis von geografischen Strukturen für Sichtbarkeitsberechnungen, Höhenbeurteilung und Entfernungsmessung. Immersive, virtuelle, geografische Umgebungen (VGEs) bieten vorteilhafte Perspektiven für eine rasche Entscheidungsfindung und Analyse von Geländestrukturen. 2D-Monitore liefern einen 2D-Eindruck einer reduzierten 3D-Umgebung, während immersive Head-Mounted Displays (HMDs) Tiefeninformationen transportieren. Darüber hinaus unterstützen digitale Planungstools die Zusammenarbeit über die Ferne durch die Bereitstellung eines virtuellen Arbeitsraums für den Planungsprozess. Ein virtueller Arbeitsraum erlaubt die Speicherung von Planungsarbeit, Visualisierung eines gemeinsamen mentalen Konzepts der Planarbeit und erhöhtes Benutzerengagement.

In dieser Arbeit untersuchen wir, wie die Vorteile immersiver, virtueller Umgebungen für die taktische Planung von Missionen und Entscheidungsfindung genutzt, und die Einschränkungen aktueller analoger und digitaler Planungswerkzeuge überwunden werden können. Wir konzipieren und implementieren eine kollaborative Umgebung in der virtuellen Realität (VR) basierend auf Anforderungen, die wir aus Beobachtungen einer Militärstabsübung und Durchführung von unstrukturierten Interviews mit Fachberatern des österreichischen Instituts für militärisches Geowesen (IMG) abgeleitet haben.

Der zentrale Beitrag dieser Arbeit ist der Entwurf, die Implementierung und die Evaluierung einer VR Anwendung, welches die Entscheidungsfindung und taktische Planungsprozesse in einer Militärstabsübung unterstützt. Die Entwicklung unseres Prototyps stellt eine Fallstudie für vergleichbare Planungsprozesse in anderen Domänen dar, beispielsweise Weltraummissionen oder Katastrophenschutzmanagement.

# Abstract


Planning tactical and strategical military operations is a well-structured and complex process and involves interdisciplinary expertise. A crucial part of the decision-making process of planning tasks that contains geographical data, is familiarizing professionals with the terrain and surrounding infrastructure. This is commonly operated by analog planning tools such as terrain models, sand tables, and standard 2D paper maps. The shortcoming of traditional equipment for tactical analysis is the lack of intuitive transfer of spatial relationships and geographical structures for visibility tasks and height judgment of ground elements. Immersive Virtual Geographical Environments (VGEs) provide advantageous perspectives for rapid decision-making and reasoning in spatial structures. 2D displays offer a 2D impression of a reduced 3D environment, while immersive displays transfer true depth information. Further, digital planning tools support remote collaboration using a virtual task space for the planning process. A virtual task space has several benefits compared to analog equivalents, such as the option to save planning states, visualize a common mental concept, no physical boundaries, and increased engagement.

In this thesis, we set out to investigate how to utilize the benefits of immersive virtual spaces for tactical planning and decision-making in the context of a military staff exercise, to overcome the limitations of current analog and 2D digital planning tools. We design and implement a collaborative Virtual Reality (VR) prototype based on requirements derived from observations of an on-site military staff training and unstructured interviews with consultants from the Austrian Institute for Military Geography (IMG).

The key contribution of this thesis is the design, implementation, and evaluation of a VR system that supports the decision-making and mission planning of officers in training and commanders during military staff training. The development of our prototype represents a case study for comparable planning tasks in other domains, such as space missions or disaster prevention management.

# Contents


**Kurzfassung** **ix**

**Abstract** **xi**

**Contents** **xiii**

**1 Introduction** **1**
1.1 Motivation . . . . . 1
1.2 Problem Statement . . . . . 2
1.3 Aim of the Work . . . . . 3
1.4 Structure of the Thesis . . . . . 4
1.5 Publications . . . . . 5

**2 Related Work** **7**
2.1 Geographical Environments . . . . . 7
2.1.1 Non-immersive Environments . . . . . 7
2.1.2 Immersive Environments . . . . . 9
2.1.3 Common Architectures . . . . . 10
2.2 Designing Immersive Virtual Environments . . . . . 11
2.2.1 The Immersive Factor . . . . . 11
2.2.2 3D User Interactions . . . . . 11
2.3 Collaborative Virtual Environments . . . . . 13
2.3.1 Presence and Communication . . . . . 14
2.3.2 Multi-modal System Architecture . . . . . 14
2.3.3 Interaction Design . . . . . 16
2.4 Digital Tools for Decision-Making . . . . . 18
2.4.1 Military Staff Training . . . . . 19

**3 Conceptual Design** **23**
3.1 Case Study: Military Staff Exercise . . . . . 23
3.1.1 Field Observation . . . . . 23
3.1.2 Derived Use Cases . . . . . 25
3.1.3 Requirements . . . . . 26
3.2 System Overview . . . . . 28

3.2.1 System Devices . . . . . . . . . . . . . . . . . . . . . . . . . 29
3.3 User Interface and Interaction Design Concept . . . . . . . . . . . . . . 31
3.3.1 General Guidelines and Considerations . . . . . . . . . . . . . . . 31
3.3.2 Design Concept 2D User Interface . . . . . . . . . . . . . . . . 32
3.3.3 Spatial User Interaction Design . . . . . . . . . . . . . . . . . 33

**4 Implementation** **37**
4.1 System Architecture . . . . . . . . . . . . . . . . . . . . . . . . 37
4.1.1 Framework and Libraries . . . . . . . . . . . . . . . . . . . . 38
4.1.2 Software Design . . . . . . . . . . . . . . . . . . . . . . . . 39
4.1.3 Data Management . . . . . . . . . . . . . . . . . . . . . . . . 42
4.2 Desktop Interactions . . . . . . . . . . . . . . . . . . . . . . . . 48
4.2.1 Desktop User Interface . . . . . . . . . . . . . . . . . . . . . 48
4.2.2 Views . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
4.2.3 Tool Menu . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
4.3 Spatial Interactions . . . . . . . . . . . . . . . . . . . . . . . . . 55
4.3.1 Spatial Controls . . . . . . . . . . . . . . . . . . . . . . . . 55
4.3.2 3D Tools . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59

**5 Results and Evaluation** **65**
5.1 Qualitative Results . . . . . . . . . . . . . . . . . . . . . . . . . 65
5.1.1 Implemented Use Cases . . . . . . . . . . . . . . . . . . . . . 65
5.1.2 Expert Feedback . . . . . . . . . . . . . . . . . . . . . . . . 70
5.2 Technical Evaluation . . . . . . . . . . . . . . . . . . . . . . . . 72
5.2.1 Terrain Test Data . . . . . . . . . . . . . . . . . . . . . . . . 74
5.2.2 Performance Analysis . . . . . . . . . . . . . . . . . . . . . . 74
5.3 Limitations and Improvements . . . . . . . . . . . . . . . . . . . . 74
5.3.1 Occlusion and Visual Clutter . . . . . . . . . . . . . . . . . . 74
5.3.2 Data Management . . . . . . . . . . . . . . . . . . . . . . . . 76

**6 Conclusion and Future Work** **77**

**List of Figures** **79**

**List of Tables** **83**

**Acronyms** **85**

**Bibliography** **87**

CHAPTER 1

# Introduction

This thesis presents the design, implementation, and evaluation of a collaborative Virtual Reality (VR) system aiming to support decision-making and tactical mission planning during military staff exercises. We want to leverage the potential of Head-Mounted Displays (HMDs) for tactical planning tasks by enabling the advantages of immersive systems and collaborative user experiences. The following sections outline the motivation of the study, state the problem to be solved, and explain the main objectives of this work, followed by introducing the structure of the thesis.

## 1.1 Motivation

Morton Heilig published a patent of the first concept of a HMD in 1960. The headset promised 3D photographic slides, stereo sound and even smell [Hei60]. Him followed Comeau and Bryan in 1961, creating a magnetic tracked HMD, called *Headsight*, with one video screen for each eye [CB61], and Ivan Sutherland proposing a mechanically tracked headset, the *The Sword of Damocles*, in 1965 [Sut65]. The ideas and visions of scientists in the years to follow empower the immersive technologies we know today and built the foundation of wearable computers and sensors essential for Virtual Reality (VR) and Augmented Reality (AR). The continuous improvement of HMDs during the last years played a major role for the widespread adaptation of the technology. Today, HMDs are used for training and education [RMFW20], emergency and disaster preparation [SD19] [HCM19] [KSD+19], design [Wol19] [BV17], therapy and rehabilitation [EMM20] [FMR+18] [GFDC20], neuroscience [RWM19], data exploration, and many more. Past research has repeatedly shown that the use of immersive virtual environments can lead to numerous benefits, for instance: the increased motivation and engagement in educational applications, [PWB+15] [ACY18] [MBGM19] [MOM19] a safe space for medical or emergency training at the speed of the trainee [KSP+19], and the increase in social skills [HG20].

A well-known merit of immersive environments is the improved spatial understanding due to a higher sense of presence, 3D perception, and the level of realism while fully immersed [BM07]. One type of application is the combination of VR systems and Geographic Information Systems (GIS), known as VRGIS, and trace back to the mid 1990. VRGIS simulate physical processes and real-life environments in an immersive virtual environment [Hak02]. The interest in immersive systems peaked in the mid-1990, and one reason was the introduction of the Virtual Reality Modeling Language (VRML), a standard file format for 3D files [BPP95]. The standard gave rise to VRGIS applications for several use cases, ranging from emergency training to education. Fast forward to 2016, the launch of Google Earth VR rekindled the interest in immersive geographical environments. Affordable HMDs just entered the market and content in VR was scare. Google Earth VR allowed users to visit any place on earth, change their viewing perspective on demand, navigate through cites using a first-person street view or explore the crowdsourced library of images and videos provided by Google's community [VR21] [KPG+17].

In the present study, we set out to investigate how to utilize the benefits of geographical environments in VR for decision-making and mission planning in a military staff exercise. We design and implement a collaborative VRGIS prototype and present a case study for similar digital support tools for collaborative decision-making.

## 1.2 Problem Statement

The term *mission* as a noun is defined by its objective and a plan how to reach it. A group of people with different expertise work on a plan before or during a mission to ensure its success. A staff exercise, for instance, a military staff training, is a safe method to plan and exercise missions. Some missions involve on-site operations as part of their procedure, such as missions for peacekeeping, rescue services, humanitarian operations, exploration,

Figure 1.1: **Right:** Example scene of a military mission training process. Source: Österreichisches Bundesheer/Rene Auer **Left:** Representation of a standard 2D paper map used during mission planning and presentation.

defense, and sometimes survival. A crucial part of on-site missions is to familiarize involved experts and decision-makers with the terrain and surrounding infrastructure of the mission site before mission execution. For this purpose, widely used analog tools for military staff exercises are terrain models, sand tables, and standard paper maps. The left image in Figure 1.1 depicts a scene during an Austrian military staff training, and the right image in Figure 1.1 shows how a 2D map is used for the planning process of the training. Overlaid transparent sheets are used to annotate important points, areas, or paths of interest. Digital alternatives of paper maps are GISs. A 3D model in a GIS consists of a geo-referenced terrain and digital imagery mapped onto the terrain data. Analog or digital maps help experts plan tactics and logistics in the field off-site, and mission leaders make rapid decisions about operational maneuvers. One disadvantage of current tools visualizing geographical information in 2D is the lack of intuitive transport of spatial relationships and vertical dimensions. Most digital 2D maps do not indicate height information without user engagement. Other spatial information essential for decision-making, such as line-of-sight, elevation, or distances, are derived using time-consuming calculations on topographic maps [PT89] [AG00]. 3D cues, as elevated maps or sand tables, enhance user engagement, reduce workload [BRS+19] [HCR+19] and improve spatial understanding of complex structures [HŠM+19] [SK03]. One evident quality of immersive technologies is 3D perception. A regular 2D display offers a 2D impression of a reduced 3D environment, while immersive displays transport depth information. Cummings and Bailenson suggest that higher immersion leads to a more realistic perspective on a given 3D space [CB16]. Further studies comparing geographical maps in immersive VR with a regular desktop monitor showed beneficial qualities of immersive settings, as increased efficiency and focused due to the sense of presence in the virtual environment [DYLM20].

Mission planning is a group activity and mainly involves the participation of experts with different technical knowledge. Digital planning tools support remote collaboration by using a virtual task space for a given planning task. A virtual task space has several benefits compared to traditional in-person collaboration, as the option to save planning states, visualize a common mental concept, flexibility concerning the design of the environment or the object of interest, and increased engagement [PA20]. Due to the recent advances of immersive and affordable HMDs, research about immersive collaborative environments is ongoing. The proposed prototype aims to overcome the limitations of 2D maps on paper or screen by using the benefits of collaborative immersive geographical environments for mission planning and decision-making.

## 1.3 Aim of the Work

The objective of this thesis is to design and develop a prototype that provides a collaborative task space in VR supporting the decision-making process and mission planning during military staff training. Within our framework, multiple users can interact with an imported 3D terrain in an immersive virtual environment from different perspectives for terrain exploration. Our application supports decision-making by offering features for bi-directional communication, awareness cues, terrain annotations, distance measure, visibility analysis, and data export in a standardized format. The application is used as a prototype in different departments of the Austrian military and will be integrated into the training curriculum for officer candidates for testing purposes. The adoption of the prototype into the training and education of officers enables the development of future digital planning tools and the identification of benefits of immersive digital tools for mission planning and decision-making. Preliminary feedback showed that the collaborative aspect, immersive viewpoint, 3D interactions with the terrain, and line-of-sight analysis provide more qualitative advantages than alternative desktop applications.

The key contribution of this work is the implementation of a VR system that supports decision-making and mission planning in a military staff training. The design of the prototype is tailored to the planning process of a military case study. Nevertheless, we argue that our design could is adaptive for other planning processes; for example, it could be used for disaster prevention and space missions. The development process is divided into three parts: First, we conduct a requirement analysis, including unstructured interviews and observations of a real-world military staff training, then we design the system and user interactions, and at last, implement and build the software.

## 1.4 Structure of the Thesis

**Chapter 2** presents state-of-the-art research and projects about virtual geographical environments using immersive and non-immersive technologies, collaborative environments, design of collaborative applications, and spatial user interactions. Further, we introduce related work similar to our use case, staff training, and tools to support decision-making and planning.

In **chapter 3** we describe the design process of the prototype. The base of the design is an in-depth analysis of the requirements, derived from observing a military staff exercise, see section 3.1. Section 3.2 specifies the composition of the hardware setup and section 3.3 presents the design of the user interactions in 2D and 3D.

**Chapter 4** is dedicated to the process of putting the system design into practice. We explain the development process and milestones, the software architecture, and used libraries in section 4.1. Essential components, like data import and export and geographical data processing is discussed in section 4.1.3. Section 4.2 and 4.3 explain the implementation and algorithms of the main terrain and collaboration tools for the desktop and the VR user and how we overcame challenges during the development process.

In **chapter 5**, the results of the implementation are shown, and insights on the technical outcome of the prototype are presented. Furthermore, we leave a critical review of the resulting prototype based on an expert interview and describe merits and limitations.

Last, **chapter 6** wraps up the most important outcome and presents improvements for future developments.

## 1.5 Publications

Partial results of this work have been published previously by the author. The following peer-reviewed publication describes the main components of this work:

B. Schlager, D. Stoll, K. Krösl and A. Fuhrmann, "Tactical and Strategical Analysis in Virtual Geographical Environments," 2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), 2021, pp. 621-622, DOI: 10.1109/VRW52623.2021.00193.

# CHAPTER 2

# Related Work

This chapter provides a detailed overview of the topics related to this thesis. These research works are within the following four domains: *virtual geographical environments*, *3D user interaction design*, *collaborative systems*, and *decision-making*.

## 2.1 Geographical Environments

The origin of any geographical information is *Cartography*. It is the study of constructing visual representations of geographical areas, resulting in artistic or topologically correct maps. Such maps are often augmented with political, cultural, or geological information [Mer96]. Maps are utilized for any use case related to the information it conveys in the referenced area, such as navigation, statistical analysis, exploration, and planning.

### 2.1.1 Non-immersive Environments

A Geographic Information System, GIS, is the digital representation of a map. The term was first introduced in 1966 during a collaboration between the Canadian Government and IBM. The goal was to keep track of the natural resources of Canada using digital maps [Goo12]. In general, GISs allow users to process, visualize, analyze, interact and quest geographic information over time using a base map and additional information layers on top, related to the position of the data on the map. The objective is to communicate the relationship between the geographical and abstract data. Abstract data is uniquely referenced using different map projections and geographic coordinate systems. A coordinate system of a digital environment follows standards from cartography, for instance, the World Geodetic System (WGS). The WGS assumes several geometric properties of our planet and offers a standardized coordinate system for reliable calculations of absolute positions on earth. The latest version of the WGS standard was introduced in 1984 and is commonly called WGS84 [Kum88].

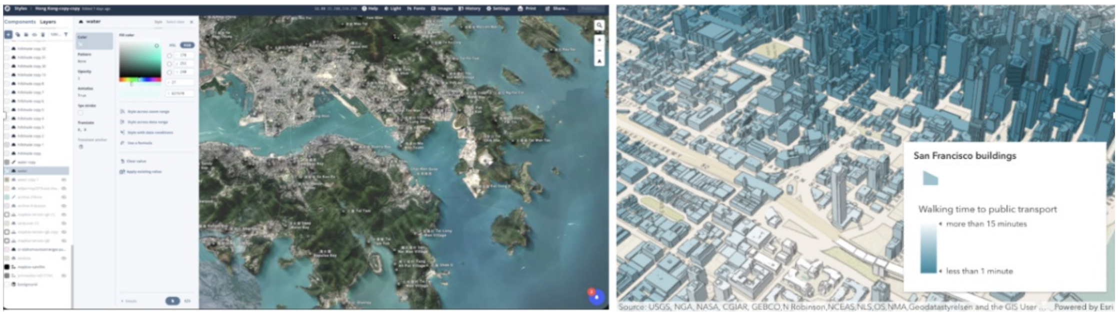


Figure 2.1: An Example of proprietary GIS software, **Left:** Mapbox Studio, a framework for creating custom maps [Map21], **Right:** ArcGIS - the 3D map indicates the walking distance to the nearest public transport in San Francisco using color code [Esr21].

There is a broad range of different GIS software, libraries, and services available. They differ in structure [RSMP11][Goo05][SB09a] and other characteristics, for instance:

- **hosting abilities** - web-based, cloud-based, stand-alone, library integration
- **data resolution** - geographical information as vector or raster data
- **interactivity** - selection and manipulation of data, dynamic loading, export
- **different images** - satellite, (unmanned) aerial vehicle imagery, photogrammetry
- **coordinate system** - type of map projection, universal grid systems
- **licensing** - commercial, free, open source
- **visualization technique** - different terrain rendering, visualization of abstract information, realistic rendering

The popular commercial GIS software vendors are Esri [Esr21] and Autodesk [Aut21]. The image illustrated on the right in Figure 2.1 shows one use case of ArcGIS, by Esri. The color gradient on the buildings in the 3D map indicates the walking distance to the public transport in San Francisco. There are open-source software alternatives to Esri available, such as QGIS [QGI21] and gvSIG [gvS21]. Another well-known provider, focused on web-based GIS solutions, is Mapbox [Map21]. The left-hand-side image in Figure 2.1 depicts an example of a 3D map created by Mapbox Studio.

In the context of GIS, the rapid development of different technologies in the early years of the 21st century enabled the combination of virtual environments focused on collaboration and geographical environments, called Virtual Geographical Environments (VGEs) [CL18]. VGEs offer a *sense of presence* and enhanced *interaction* techniques with the data using avatars and allow unique perspectives. Compared to regular Virtual Environments (VEs), VGEs use data derived from the surface of the earth. VEs immerse users in any environment, as grounds from other planets or crafted cities using the creators imagination [CL18].

The main difference between GISs and VGEs is the presentation of the data and the role of the user. VGEs provide perspectives and views in the environment, primarily in 3D.

Investigating how we understand maps is as complex as any stimuli. Users often rely on depth cues communicated through shading or surface topologies to derive information. This is an advantage of 3D maps, in contrast to systems using 2D rendering [BRS+19]. Schmidt-Daly et al. [SDRH+16] analyzed different tools for spatial knowledge acquisition and spatial reasoning skills. In their study, three experimental displays were compared, a paper map, a digital 2D map, and a 2D map projection on a sand table. They observed that participants that used the augmented reality sand table achieved better results for landmark identification and distance estimation tasks. Another evident advantage of 3D maps is the visualization of vertical information of data points. The visibility of the third dimension allows us to derive a relative comparison of terrain data.

### 2.1.2 Immersive Environments

A well-known medium for immersive VGEs are Cave Automatic Virtual Environments (CAVE) [CNSD+92]. A CAVE system creates the illusion of depth and immersion by projecting the virtual space on a room-sized cube with more than three walls. Users perceive depth information through stereoscopic glasses, tracked by the system to adjust the renderings to the viewing pose of the user. CAVE systems are used for numerous use cases; examples include: product design in the automotive industry [Zim08], exploration of geographical data [SB07], education [PWB+15] and data analysis for disaster prevention [HCM19]. Past studies compared immersive CAVEs with a standard screen and run common tasks, like spatial comparison and search. The high-immersion setup provided more accurate results and improved the efficiency for the given tasks due to stereo perception [SB07] [PWB+15].

The combination of GIS and VR technology, VRGIS, provides similar benefits. In contrast to CAVEs, users only perceive the virtual environment through a HMD and do not see the external world around them. This leads to higher immersion and a sense of presence

Figure 2.2: Two examples of immersive virtual geographical environments. **Left:** A CAVE with three walls [SB07], **Right:** A person using a state-of-the-art VR HMD [CMH16]

in the virtual space. Figure 2.2 shows one example of a CAVE system and a person using a VR headset interacting with a geographical environment. Previous studies showed that VRGIS improve the spatial understanding and perception of structures with complex geometry, as opposed to standard desktop monitors [SK03] [HŠM+19].

Other HMDs used for visualizing geographical information are AR displays. Users can see localized virtual information on top of the real world. AR displays are preferred if a use case aims to display additional information of a specific location on-site. Most AR headsets reconstruct the user's environment, track the user's position over time for precise localization, use additional sensors and cameras to process the user's input, and augments the real environment with virtual information. Visualizing geographical terrain not related to the location of the user is beneficial for limited, specific use cases; for instance: displaying underground pipeline systems [SMK+09] [ZHHL16] or urban infrastructure and constructions on-site [CKW13].

### 2.1.3 Common Architectures

Since the introduction of GIS, numerous different types of GIS architectures have been published. Some run as stand-alone applications and use open-source data, like QGIS [QGI21], and others are web-based, accessible from any web browser [AG17]. Current literature reference stand-alone GIS applications as *desktop GIS*. Desktop GIS manages, views and stores the data locally. Some local GIS software queries geographical data from an external server using GIS libraries and extensions [SB09b]. In contrast to desktop GIS, web-based GIS, also called *webGIS*, queries the data through a browser. One of the most utilized examples for webGIS is Google Earth [Ear21] and OpenStreetMap [Ope21]. There are several webGIS software architecture types: client-server architecture, service-oriented architecture and cloud computing. Each type differs in scaling capabilities, data management, security sensitivity and workload on the client. Chen et al. [CLL17] proposed a high-level conceptual software architecture for VGEs consisting of four sub-environments: data, modeling/simulation, collaborative and interactive component. Each environment presents an important character of the system: *Data environment*, as the main component holding spatiotemporal information, topological relationships and in some cases further semantics, like evolutionary processes. *Modeling and simulation environment*, information derived from geographical information. *Expression environment*, concerned with visualizing the information of the system and *collaborative and interactive environment*, representing the tools for human interaction with the geospatial data. According to this system concept, VRGIS utilizes VR to represent the expressive and collaborative/interactive environment.

This thesis focus on the design and development of a VRGIS using a desktop-based architecture. The Institute for Military Geography (IMG) in Austria, provides us with prepared geographical data. Similar to desktop GIS, this work uses specialized libraries to import the local data into the immersive environment. Chapter 3 describes this process and the architecture in detail.

## 2.2 Designing Immersive Virtual Environments

This section presents an overview of general design concepts and state-of-the-art 3D user interaction metaphors for VR spaces.

### 2.2.1 The Immersive Factor

The main advantage of a VR headset is the degree of immersion it provides to its users. According to Mel Slater [Sla03], *immersion* is the level of sensors delivered by a system and can be quantified by the number of stimulating sensors, like auditory, haptic, display technology, tracking, latency, and many more. It is linked to the plausibility of the virtual environment and influences the sense of user presence. *Presence* is the response of the user to the immersive system and is described by the sense of being in the artificial environment. The sense of presence is highly subjective and varies over time while using an immersive system [BM07]. Immersion and presence are linked to each other but represent different cognitive concepts. Another important cognitive sensation in VR is *user embodiment* and defines the degree of awareness of the user's own virtual body. Kilteni and Slater [KGS12] describe embodiment as a combination of three sub-components: self-location in space, sense of agency over the virtual body, and sense of body ownership. Previous studies showed promising results for pain therapy [MGDB+19] and empathetic engagement [BGR+18] when altering the user's virtual body in VR.

One unfavorable effect of immersive headsets is *simulator sickness*, also called cyber-sickness, and refers to a group of physical and emotional symptoms experienced during or after using a VR system [KLBL93]. Symptoms include dizziness, nausea, headaches, eye strain and disorientation [LJKM+17] [KSD00] [CKY20]. The exact cause for user discomfort is unclear but past research proposed several factors influencing VR sickness. Known influential parameters are exposure time, technical specifications of the HMD, as display refresh rate, physical characteristics, like age of the user, used locomotion techniques, and the design of the content in virtual environments [SSB+20].

### 2.2.2 3D User Interactions

An essential characteristic of VR technology is the user interactivity in 3D spaces. Users engage with the virtual environment and receive multi-sensory feedback. 3D user interaction design describes techniques for efficient Human Computer Interaction (HCI) design in 3D environments and has three main objectives [Kru12] [LJKM+17]: usefulness, usability and emotional satisfaction. Usability defines the user's comfort and ease of use of the system itself, while usefulness determines the system's capacity of support for the task goal of the user. The emotional satisfaction of an immersive system depends on several factors, like trustworthiness, performance, and aesthetics. Universal interaction metaphors striving to achieve the objectives of HCI in 3D environments are divided in four categories: *navigation* which is defined as viewpoint control, *selection*, *manipulation* and *system control* [LJKM+17].

### Selection and Manipulation

Manipulation tasks in a virtual space include selection, positioning, rotation, and scaling of virtual objects. They can be further broken down into 3D interaction metaphors such as *grasping* with a virtual hand, *pointing*, like ray-casting, selection, and manipulation of *surfaces*, for instance, 2D multi-touch surfaces, *indirect* selection, and manipulation, *bimanual* (using two hands for interactions), and hybrid selection [LJKM+17]. HCI research proposes building blocks for selection and manipulation techniques. The former consists of a method of indication, like touching or pointing, then confirmation of the selection, and finally, the feedback. The confirmation of the selection and the feedback can be communicated to the user visually, through voice, gesture, or text. The latter, manipulation, consists of similar components: object attachment, object position, object orientation, and feedback [BJH01].

### Navigation

While traveling and wayfinding through our physical reality are trivial, traveling through a virtual environment is not. The challenge is to design an intuitive navigation technique depending on the use case and task. Common reasons to move from one place to the other are exploration, search, or maneuvering. Generally, a travel task can be divided into three components: start to move, indicating a position, indicating an orientation, and ending the movement with a final position [BDHB99]. State-of-the-art locomotion techniques are metaphors for *steering*, directed by gaze or orientation of an input device, *target-based*, choosing a specific location, *route-planning*, defining points over time, and *direct manipulation* of the view point [LJKM+17]. Traveling methods in VR, known as walking metaphors, are redirected walking, real walking, granted that the virtual space corresponds with the real environment, and treadmills [LJKM+17].

As seen in Figure 2.3, Google Earth VR [VR21], uses a variety of navigation metaphors. Users can search for a specific location via text input and, upon selection, are transported to the desired place. On the other hand, users who want to travel short distances on the

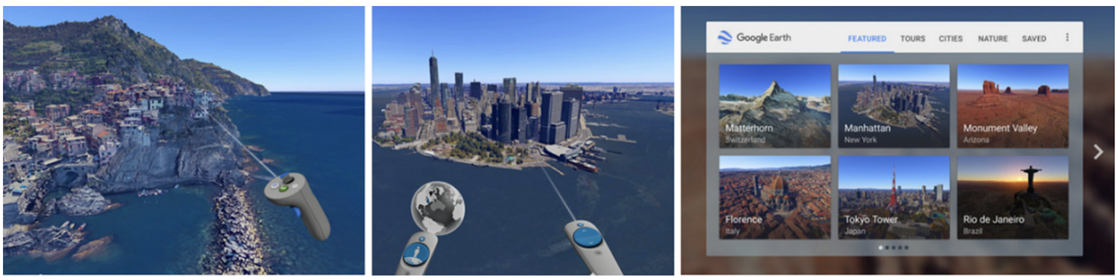


Figure 2.3: Different navigation modes in Google Earth VR. Images courtesy of [iGEV21]. **Left:** Teleportation using ray-casting. **Center:** Users utilize the world map in the left hand to choose a location. **Right:** Text-based search and travel.

map either teleport to a nearby point using a ray-cast or directly steer to the desired location, using the joystick buttons on the input device [KPG+17].

### System Control

System control tasks are related to regular commands on 2D User Interface (UI)s (UI) and often define *what* should take effect. System commands request or change a specific action or change the system state entirely [LJKM+17]. 3D menu design criteria are categorized in [DH07]:

- **size and number:** menu system or single menu
- **appearance and structure:** text-based, geometric structures,
- **input setting:** using hand tracking, or other tracked devices.
- **availability:** temporary, fixed or user dependent
- **position and orientation:** spatial (world-based) vs. diegetic (integrated in the 3D world)
- **compatibility and usability**

One common menu design for system control is directly translated from a 2D interfaces: *floating menus* [DH07]. Users select items on a 3D surface in the virtual space using ray-casting or direct touching. Other types of menus regularly used are geometric or symbolic structures attached to the body or the controller of the user, for example, a wristwatch or ring menu [SZDA17]. Similar to other interaction techniques, there is no universal menu design suitable for every type of virtual environment. LaViola et al. [LJKM+17] propose general design guidelines for 3D interactions. One suggestion concerns visibility; users should not have to search for commands for state changes. Novel input metaphors, as voice or gesture, should be introduced and properly adapted to the overall function of the application. The authors also mentioned that 3D UIs might not benefit every use case and propose hybrid interfaces.

3D interaction design highly depends on the 3D composition of the virtual space, using input devices and use cases. Current literature proposes different design patterns for specific cases; for instance, if direct, realistic manipulation of an object is necessary, using a selection metaphor utilizing hand tracking is advantageous [Jer15]. Chapter 3 offers a detailed overview of how we incorporated state-of-the-art design guidelines for spatial interactions in the context of collaborative decision-making for mission planning in VR.

## 2.3 Collaborative Virtual Environments

Collaborative Virtual Environments (CVEs), compared to single-user virtual environments, offer additional features for social interactions. The objective of CVEs is to transform a digital environment into a rich 3D space in which multiple users can interactively navigate, communicate and share a given context [BGRP01, SAT21]. Characteristic features include: choice of viewpoint and movement in the virtual space, access for multiple users simultaneously, different types of input for communication, as text or voice, user embodiment, mutual interactions with virtual objects, different meeting and interaction scenarios between users, adaptive broadcast of information and balance of power, like an active speaker or listener [GB95, SCM01]. A CVE is a technology that can significantly improve some aspects of social interactions not realizable by audio and video conferences alone. Access to an immersive virtual 3D space has a positive impact on peripheral awareness and allows sharing of resources and content uniquely in a 3D space over time [LPBR17, SS15].

### 2.3.1 Presence and Communication

A central feature of a collaborative space is communication, more precisely, exchanging information between participants in a shared space [CS98]. Users communicate with others in immersive virtual environments either through direct channels, as audio input and output, or through non-verbal communication cues, using their surrounding space, with gestures, body language, and facial expressions [TZS+18].

One common cognitive sensation in an immersive virtual space is the sense of presence, which was introduced in section 2.2. While presence refers to the feeling of being in the virtual environment, the sense of being in a virtual space in company with others is called *copresence* [Zha03]. Copresence is related to *social presence*, the degree of being able to interact with other people in the environment and form a relationship. The effect of copresence in virtual environments is yet uncertain due to different results in past studies, but some research suggests that social presence influences user satisfaction, enjoyment [Bul12] and trust [HH07] in others. Several factors influence social presence: the technological character of the system, for instance, the type of haptic feedback and depth cues, and visual presentation of user behavior, and the design of the user avatars [OBW18]. An avatar is a virtual portrayal of a user in a virtual environment. Avatars differ in visual representation, for instance, in their level of detail, head-only or full-body, and realism, a comic-like or realistic design. Further, Steed and Schroeder [SS15] categorize avatars in three types according to user movement: puppeteered, as controlled by the user, tracked by sensors, and reconstructed, like real-time 3D video capture. The type of collaboration in the virtual environment usually defines the necessary level of detail of user avatars. In task-driven CVEs, realistic and detail-rich avatars might be non-essential, but virtual environments focused on social interactions and communication benefit from accurate tracking of facial features of users [ND14].

### 2.3.2 Multi-modal System Architecture

Numerous applications areas require access for multiple users in virtual environments. CVE are utilized for social applications, like education [BYB+08, GKK+17, GCM+19], entertainment [MSMSMI18], training [WWL+19, SMR+18, SRD15], engineering [MWA+19, MMB20], data analysis [HCM19] and even remote usability testing [MG17]. The type of collaboration varies depending on the use case or used hardware and can be categorized in time, synchronous vs. asynchronous, and space, remote vs. on-site collabora-

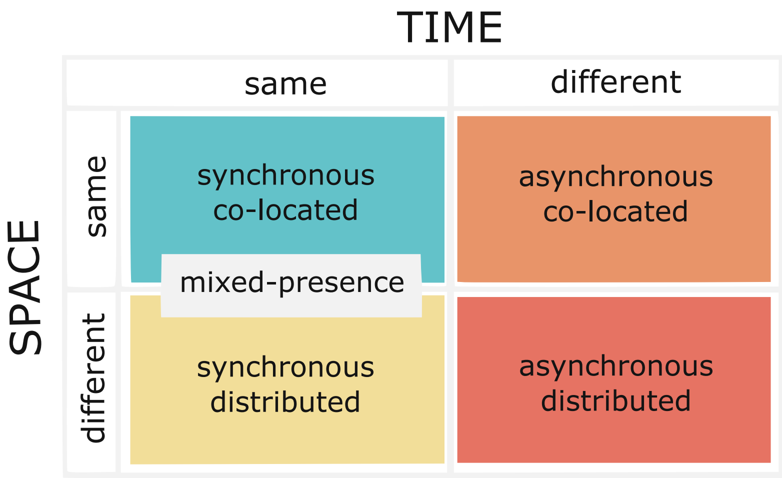


Figure 2.4: Space and time taxonomy of collaborative systems [NLSB19] [TBG05]

tion [NLSB19], see Figure 2.4. Current literature uses the term on-site and co-located, as well as remote and distributed, interchangeably [LPBR17]. Most collaborative setups are synchronous; that is, involved parties work at the same time. As seen in Figure 2.5 the difference between synchronous on-site and synchronous remote is the physical location of users. The on-site collaboration uses the same physical space to engage in a virtual task, while users involved in remote collaboration do not share the same physical location.

### Synchronous On-Site Collaboration

Users in an on-site CVE share the same physical environment at the same time. A physical environment can either contain a tracked space or several disconnected tracked spaces. Most use cases utilizing this type of collaboration assume that the virtual space has the same extent as the physical one and is registered to the real environment and objects. All users navigate together through the virtual space; otherwise, the virtual position of the avatars is out of sync with the body of the actual users. The safety in the shared, real workspace has to be secured for all users involved in the virtual space to avoid collisions. Lacoche et al. [LPBR17] proposed several techniques to allow multiple users to navigate independently in a virtual space while being in the same room. Additionally to a regular avatar in the virtual world, the authors propose different visual metaphors to indicate the pose of the other user in the real space. Another method to avoid collisions in large tracked spaces is redirected walking [AGR17] [PVS+16]. Redirected walking methods rotate the virtual space, unnoticed to the user, to create the illusion of infinite physical space.

Co-located collaborations not only differ in the type of tracked space, but also in the type of display device used. In symmetric VR/AR, all users wear either a VR headset or a AR headset. A setup where non-HMD and HMD users, as well as AR HMD and VR HMD, work together, is called asymmetric [GDM19] [GSFR17]. Users without

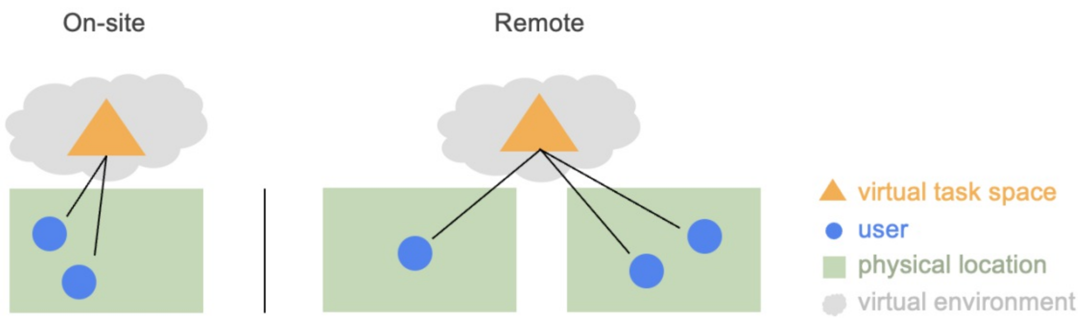


Figure 2.5: An abstract setup of a remote and on-site immersive collaborative virtual environment. The difference is the physical location of the users utilizing the virtual environment.

headset usually have access to the virtual environment through a regular 2D medium and interact with objects in the virtual environment through a touch sensitive display or peripherals [DF09] [GDM19] [GSFR17]. Asymmetric collaboration can be used for data analysis tasks [SPB+20], gaming [SCG21], architectural design [SIC+18] and others.

#### Synchronous Remote Collaboration

Common use cases for remote collaboration are social VR [LVW+21], remote guidance [MG17, ESO+17], communication [SS21] [ELL+18] and training. The interest in social VR has increased steadily over the last years, due to the decrease in costs for consumer VR headsets. AltspaceVR [Mic21b] and VRChat [VRC21] host virtual spaces to communicate with strangers in public, or connect with friends in private rooms. On the other hand, SpatialVR [SS21] is customized for business use cases, like remote meetings and collaborations. SpatialVR additionally supports the use of different hardware at the same time. Users can connect to a remote meeting room using their VR, AR headset or 2D desktop.

#### Asynchronous Collaboration

In applications designed for asynchronous collaboration, users collaborate and communicate at different times. Users can not communicate via voice or gestures in real-time but share text, notes, sketches, or other annotations. This type of collaboration is mainly used for specialized applications, for instance, annotating a 3D urban design process [KD10] or teaching and learning in a virtual, collaborative classroom [MGFS13].

### 2.3.3 Interaction Design

Collaborative environments demand specialized interaction methods for communication with other users and engagement with virtual objects. Otto et al. [ORW06] suggested that several characteristics of CVEs influence the effectiveness of closely-coupled collaborations: Technological factors, as immersion and field-of-view, human factors, as presence and manipulation techniques and application factors, as usability and task design. This section presents state-of-the-art research for human and application factors.

Pinho et al. [PBF02] proposed several design considerations for manipulation techniques in CVEs [SAT21] and respective examples, as seen in Table 2.1. Visual clues, such as primitive geometries representing the shared task space, gestures, user embodiment, and user attention indicators [BKS99], allow effective conversations in CVEs and increase the awareness factor [KFS03]. Garcia et al. separated user awareness into seven sub-categories [GMMG08]: Object state awareness (selection, modification, ownership), task awareness (participants, possibilities, status), world awareness (position of objects and users, world structure), group awareness (online status, position), social awareness (communication, gestures) and system awareness (stability, latency). In the context of task awareness, Kraut et al. defined four types of visual information per collaborative sub-task [KFS03]. As an example, in case of monitoring other people's actions in a shared space, one can utilize the participants head and face (gaze direction), the participant's body and action (change in pose over time), the task object (change in pose over time) and work context (the traces of the task over time in the workspace).

Common visual cues for sharing awareness are ray-casting, visualizing or extending the gaze of the user, sharing the view frustum [PDE+19] or virtual sketching [KLH+19]. If users work virtually close, hand tracking can be further used to share attention and awareness with others [SGBB19]. Ray-casting, sometimes called pointing, is widely used to indicate attention [BKS99] and facilitate object manipulation [RHWF06]. Depending on the use case, users in immersive CVEs share a virtual task space for common interactions, like sketching or manipulating objects. There are various possibilities to arrange the task space in a virtual scene. He et al. [HRP19] compared three scenarios, a side-by-side collaboration, similar to a whiteboard, face-to-face, where the space between the users is used for annotations, and an eyes-free setup, where users can interact on a real table, instead of drawing mid-air. Participants showed different communication behavior, like increased eye contact and focus in the face-to-face setting. In general, a collaborative

| | |
|---|---|
| **Awareness** | Users should be aware of each others tasks and their performance in the environment under test |
| **Evolution** | Transfer the rich knowledge on single user techniques to multi-user case |
| **Transition** | Maintaining immersion in the virtual environment when transitioning between single-user and a collaborative task |
| **Reuse** | Novel interaction metaphors are based on existing code |

Table 2.1: Design considerations for effective manipulation techniques in virtual spaces proposed by Pinho et al. [PBF02]

task can be assigned to multiple users using *distinct attributes*. For instance, user A performs sub-task 1 and user B, sub-task 2. Another method for multi-user collaboration is using *same attributes*, user A and B work concurrently to perform sub-tasks 1 and a subsequent sub-task 2. One example of this setting is the virtual gazebo construction project [RWOS03]. In the course of the study, the authors asked users to carry a wooden beam together, as an example for a task with the same attributes, or each user was drilling or screwing by themselves, as a task with distinct attributes. Distinct and same attributes can be built by sequential (asynchronous) or concurrent (synchronous) task allocations in CVEs [ORW06]. Past literature showed that various working arrangements in a virtual space and different combinations of headsets are used successfully for many tasks.

This thesis aims to design a collaborative setup tailored to predefined requirements and match the interaction techniques to optimize communication between the users for decision-making and planning tasks.

## 2.4 Digital Tools for Decision-Making

Numerous factors influence decision-making. Some examples are vision, greater selection, probability, risk, experience, politics, emotions, and others. Some decision-makers utilize digital tools and simulations further to identify the character of a situation or problem, aiming to optimize the process concerning the outcome [OS93]. In a broad spectrum of industrial applications, there are efforts to use CVEs for decision-making. For instance, disaster management [EJAL16], emergency training [Sha20] and design [SAX+09]. The objective of these virtual environments is often to provide a comprehensive perspective on a task or to recreate a real experience and to prepare or support decision-makers[SAT21].

One process requiring consistent decision-making from involved parties is planning. Planning is the arrangement and configuration of actions towards a predefined goal. Digital tools supporting asynchronous and synchronous remote collaboration in a shared virtual space could lead to a global decrease in decision-making time and costs. Research work published by Menck et al. showed encouraging results using immersive virtual environments for factory planning [MYW+12]. The authors studied the use of VR for specific sub-tasks of a factory planning process, the so-called central planning phase. During this phase, several experts collaborate on the concept and details of the factory plan. Divisions include management, product engineering, logistics, business economics, construction workers, maintenance, suppliers, and customers. VR is used for collaborative meetings, visualization, and design. The interaction methods are adapted to the use case. For instance, for virtual meetings, the authors implemented features for social interactions, and for collaborative design, users could additionally navigate, annotate and manipulate objects in the VE.

VR technologies are used for mission planning in space since the early 1990s [KWS94]. A more recent distributed immersive system to prepare for space missions was proposed by Garcia et al. [GFR+19]. Scientists from different backgrounds use a digital environment, either an immersive display or a regular 2D monitor, for remote collaboration, and exploit potential landing sites, analyze the atmosphere of Mars, and select targets for the rover. Another domain is urban planning, Roupé showed that two main factors are influencing the efficiency of communication and decision-making in immersive environments [Rou13]: human information processing and technical aspects of the system. Human information processing incorporates cognitive processes, like reasoning and spatial perception, the background of the users, and task goals. Technical aspects include the hardware itself, display type and degree of immersion, the visual presentation of information in the virtual scene, and interaction methods. The system accelerates the decision-making process by providing a broader understanding of spatial structures and their interactions with their surroundings. The technical aspects, as well as the human information processing, should be considered when designing a collaborative tool for decision-making. Small deviations or latency concerning the portrayed information in virtual environments could lead to suboptimal results and a decrease of the benefits [DZSZ18] of the system.

### 2.4.1 Military Staff Training

Simulation devices and technologies are manufactured for military purpose since decades [Lel13]. The first immersive flight simulators were used to train future pilots [Lel13] [BH16]. A technical report by the North Atlantic Treaty Organization (NATO) reviewed countless projects related to VR in European countries, and according to the authors, VR offers military trainees an environment to learn and perform military tasks [RF03]. Mission simulations in VR allow safe practice and rehearsal prior execution [HAM+03] and therefore support decision-making [Lel13].

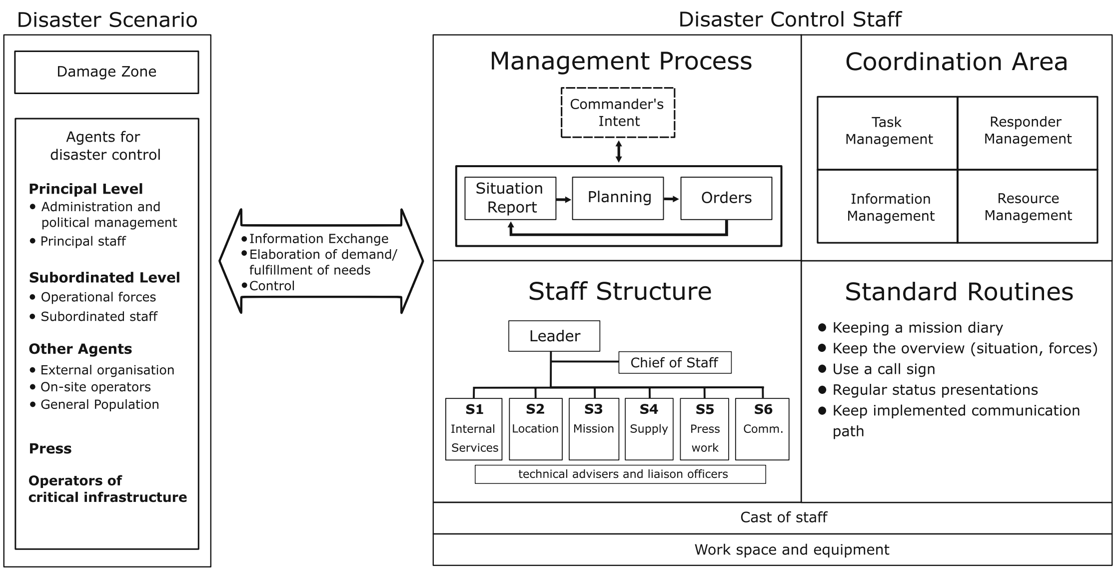


Figure 2.6: Proposal of a high-level conceptual staff exercise structure. The figure displays the relationship between the crisis scenario and crisis prevention staff. [HRL14]

A typical setting in the context of crisis management is military staff exercises. Such exercises regularly aim to prepare decision-makers and execution units for unexpected circumstances in times of crisis. Participants belong to a particular group, also called staff, and every group has a unique role, for instance, resource or communication management. In general, staffs consult the decision-maker by acquiring information about the situation within their proficiency. People in charge use the collected information to translate them into actions against high-pressure deadlines [HH16]. The goal is to systematically simulate crisis scenarios and design actions based on decisions made in an interdisciplinary environment. Further, to strengthen leadership skills, teamwork, and rapid reasoning with limited resources. Staff exercises are not exclusive to military training but are also used by other civil services, like the police, fire department, hospitals, and others. The goal is to rehearse systematic planning during unpredictable events, like natural disasters, war, or cyber attacks [HH16].

Depending on the context, the structure of a staff exercise can vary in terms of communication, scheduling, hierarchy of involved experts, and mobility [HH16]. Heumüller et al. proposed a conceptual structure of a staff exercise for crisis management, as seen in Figure 2.6. The model consists of an artificial crisis scenario, the crisis environment, and involved agents (left in Figure 2.6) and the crisis staff (right in Figure 2.6). Both parties exchange information, for instance, status reports, requests for further resources, or high-level orders. Leaders in charge define the problem space, construct actions, and evaluate the executions of staffs [HRL14]. A critical aspect of proper decision-making during staff exercises is situational awareness. Situational awareness describes a comprehensive perception of a specific situation and its correct interpretation [QG16]. High-level staff members or leaders can ask standardized questions during briefing or debriefing [QG16] or utilize other digital tools to increase situational awareness.

Digital support tools for staff exercise include technologies visualizing the crisis environment [MK16], either a GIS in 2D or 3D [AG00], as seen in Figure 2.7. In a preliminary study, Alexander et al. investigated the potential of using VR and AR headsets during the briefing and debriefing process of air force mission planning [ARD+19]. The authors

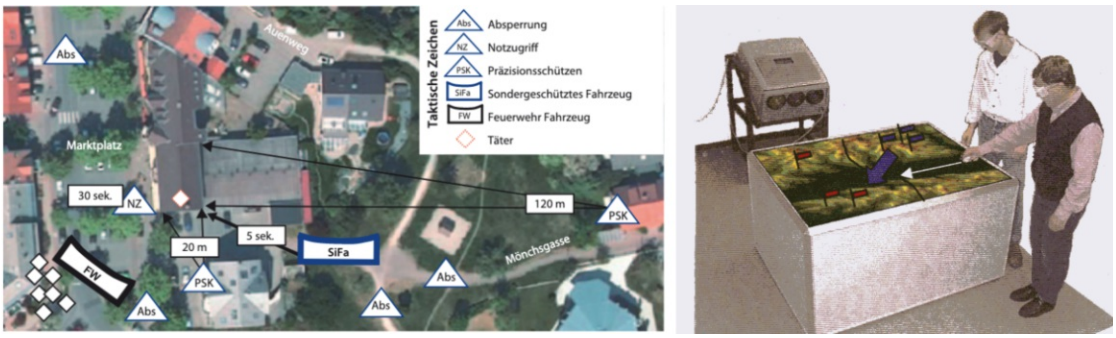


Figure 2.7: **Left:** An example of a tactical situation report in 2D space [MK16], **Right:** A digital sand table for tactical situation analysis in 3D space [AG00]

determined that immersive CVEs may support the evaluation and performance assessment of air force missions after execution.

The previous sections described research work related to the proposed prototype in detail and showed the benefits of using geographical immersive CVEs for mission planning and decision-making. Key designs and methods of the related work will base the design and implementation of this prototype. For this thesis, we participated in an Austrian military staff training for prospective officers, constructed a conceptual model of their planning process, and extracted sub-task for immersive technologies. The goal is to optimize the decision-making process of officers in charge. The next chapter describes the design of our prototype in detail.

# CHAPTER 3

# Conceptual Design

This chapter presents the requirements, the hardware and software components, and a design concept for the developed prototype's user interface and interactions. Functional and non-functional requirements are derived from literature research, consultations with the Austrian military's geographical department, and a field study at military staff training. We created a conceptual design of the VR system and interaction metaphors for collaboration and object manipulation in 3D spaces based on the requirements.

## 3.1 Case Study: Military Staff Exercise

The definition and structure of a staff exercise have been introduced in chapter 2.4. The following sections focus on military staff training and describe the conducted requirement analysis for the developed prototype in detail.

### 3.1.1 Field Observation

At the beginning of the design phase, we had the opportunity to visit a military staff training on-site and integrate the observations into the design of this prototype. The staff exercise took place in Lower Austria, specifically in the Theresian Military Academy. For more than 200 years, the academy has been training prospective officers and offers its historical rooms for military research [fLB21]. The military academy regularly hosts staff exercises for officers in training from all over the country to teach leadership, facilitate teamwork, and solve problems. Several groups, the so-called *staffs*, cooperatively work on complex tasks to support the decision-making of the commander in charge during mission planning and execution. The future officers bear extensive responsibilities during the training sessions. In some cases, hundreds of soldiers are involved, and their expertise and infrastructure vary from air, water, or land operations. The staff training can take up to eight weeks to strengthen social skills, planning proficiency, and operational competencies

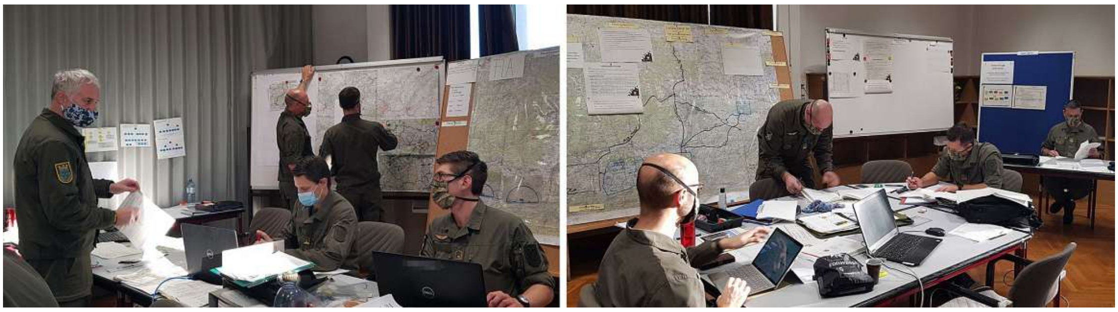

Figure 3.1: Example scene of a military staff training, Source: Österreichisches Bundesheer/Rene Auer

while managing large military units in everyday scenarios such as defense, protection, or invasion.

The field study took place at the beginning of a two-week-long staff training session and lasted three days. The goal was to passively monitor the behavior of the officers in training, their way of communication, group dynamics, and planning process. At the end of each day, we had short interviews with personnel familiar with the staff training and eager to use immersive devices to support staff exercises.

The training was executed in a large room, and the room was separated into several working areas equipped with numerous paper maps of the operational area, tables, chairs, digital support tools (e.g., printers and computers), similar to the scenes seen in Figure 3.1. Every working area was dedicated to one staff, and every staff had to solve complex problems in their field of expertise and experiences in the given fictitious scenario. While some staffs were responsible for air operations, others might plan the supply management during the mission. Figure 3.2 depicts the timeline of the observed staff exercise. The schedule continuously repeats four phases: a planning phase, followed by a briefing, mission execution, and debriefing phase. Depending on the tactical task, one cycle can take from 6 to 24 hours.

**Mission Planning:** During the planning phase, trainees work in their dedicated section of the room and utilize different tools, such as computers and paper maps, for planning. This phase aims to gather information about a given scenario and plan actions to solve a tactical task. A typical output is an annotated transparent plastic sheet pinned to a paper map of the application environment. The annotations on the sheet visualize point-of-interests, paths, or areas of opponents, allies, or other agents and describe a

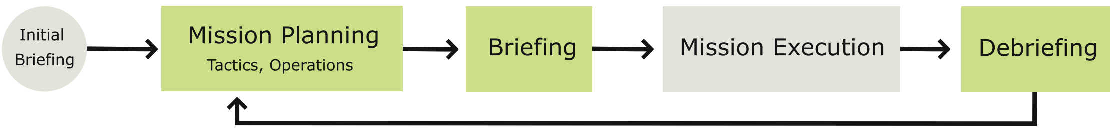


Figure 3.2: The timeline of the observed staff exercise

strategy for the given context. Part of the planning process is terrain analysis and investigation of the local situation. The terrain influences the annotations and placement of involved mission agents. Officers in training have to communicate efficiently and learn how to use their short time for planning wisely.

**Briefing:** In this phase, all groups come together and present their strategies to the other groups and the commander. The annotated transparent sheets of each group are overlaid on the paper map of the application area. The commander provides feedback, gives advice, and orders instructions to each group according to the presented plans. One challenge for the commander is to keep the general goal of the mission in mind while merging the plan and strategies of each staff. Therefore, the commander might ask the officers about terrain characteristics, visibility, and other information regarding the presented strategy. Some examples are: *"How long does operation group A need to walk to position X?"* or *"Can operation group A see point X from position Y?"*. Additionally, the commander comments on strategies according to his experience and continuously points out points-of-interests on the map, such as *"Our ally B at this position might decide to go this way."*.

**Mission Execution:** Depending on the military staff training structure, the mission execution is rehearsed by soldiers in a training environment related to the scenario or invented by agents involved in the training organization.

**Debriefing:** During debriefing, all staff and the commander in chief analyze every event during mission execution and their results. The commander discusses the successive steps and further tasks for the next mission planning iteration.

### 3.1.2 Derived Use Cases

During the planning phase, the officers in training might utilize digital maps, such as VGEs, for terrain analysis and exploration. The conducted state-of-the-art review in the previous chapter shows that immersive VGEs have the potential to further support decision-making in planning tasks. Based on the observed activities, communication between staff, and overall structure of the staff training, we extracted two main use cases: using an immersive VGE for I) terrain annotations and II) visibility and quantitative analysis.

- **Terrain Annotation:** This use case aims to support officers during the planning phase of the staff exercise. The 3D perception in immersive environments helps understand spatial structures and intuitively annotate the terrain. Drawing a line or placing point-of-interests is as easy as moving hands in the physical space. Another advantage is the collaborative feature of virtual environments. Using virtual environments, more than one user can annotate the terrain and work on the given task simultaneously. The virtual environment represents a virtual task space, similar to the paper map during mission planning.

- **Visibility and quantitative analysis:** Common tasks when using paper maps are distance estimation and line-of-sight analysis. Paper maps or 2D maps can be cumbersome for such purposes. Immersive virtual environments offer an intuitive way to analyze visibility from a single viewpoint and extract quantitative measurements, such as angles, slopes, distances, or areas.

### 3.1.3 Requirements

Based on the derived use cases, observations, and informal discussions with end-users, we identified the following requirements for the prototype:

- **Collaborative environment:** Mission planning and preparation is a multi-user task. The system should offer an interactive task space for at least two people and support bidirectional communication for symmetrical and asymmetrical collaboration. For simplicity, the first iteration of the prototype assumes two users working simultaneously in the same room. For this prototype, there is no need for remote collaboration support.

- **Desktop functionalities:** One part of the functionalities should be available to the desktop user without the need for VR hardware. Thus the application is mainly designed to be used with an HMD, users should be able to make small changes on the desktop. For instance, removing or renaming already set military annotations or placing point-of-interests. We observed that officers have to work quickly; thus, simple adjustments on the desktop are welcomed, such as tools for editing or adjusting annotations and measurements. During the planning phase, the desktop operator should have the ability to support the person using the headset by interacting with the data on the desktop and observing the work of the person in VR on the desktop screen or projections to a wall.

- **Support planning and presentation simultaneously:** The system must support terrain analysis during the planning phase, as well as during briefing and debriefing. During the planning phase, the goal is to annotate a map according to the objective and mission of the staff. During briefing/debriefing, the system is used to present a tactical strategy. Unique user goals should be considered when designing user interactions and features.

- **Geographical data import:** The application is required to enable importing a custom data package, including geographical data, metadata files, and files for military markers and annotations. Metadata files hold parameters for the geographical data, as extents and scale. The military markers and annotations' data should follow an industry standard to enable compatibility with third-party applications used by the military.

- **Geographical mapping:** One crucial property is the correct alignment and mapping of the geographical coordinate system with the 3D virtual environment.

Officers should be able to query geographical information, such as the Universal Transverse Mercator (UTM) coordinate of a position on the terrain, directional measurements, angles, or distances. The application is expected to provide such features reliably.

- **Support for multiple terrain layers:** The terrain data may include several visual layers, such as satellite images or images of urban infrastructure. The system must load and map the texture onto the terrain data correctly.

- **Support for various spatial resolutions:** Raw terrain data vary widely in magnitude, from detailed urban environments reconstructed from drone images with a range of few hundred meters to coarse elevation data, such as Digital Elevation Models (DEMs), derived from satellite images with an extent of more than 100km.

- **Annotation layers:** Annotations ought to be organized in layers, similar to the plastic films used for the staff training. The officers add annotations based on their planned strategy to their unique annotation layer. During briefing or debriefing, the annotated layers of each group are required to fuse all output into an overall mission plan. Therefore, the application must support isolated annotation layers for experts during planning as well as functionality to add additional layers and overlay them in the presentation phase. Nonetheless, it is desired that the system avoid occlusion and information clutter by allowing transparency adjustments of each layer.

- **Reusability:** Annotated maps are the main output of the VR system. The system should be able to export and import annotations into the system without loss of information and export the data in a standard file format to facilitate print-outs of the planning results.

- **Quantitative assessments:** A VR environment facilitates distance and depth estimation thanks to its stereo perception. It offers a simple interface to query distances, angles, and dimensions of spatial structures. Such quantities must be delivered as accurately as possible.

- **Usability:** The 3D user interactions and 2D interface on the desktop will follow state-of-the-art metaphors and should support users to conduct their tasks with ease and efficiency. In an ideal scenario, implemented techniques for object selection and manipulation in VR are designed for amateur as well as experienced users. Similar tasks are implemented with a similar user interaction metaphor to avoid unexpected behavior and frustration. This way, the user can return to a normal state in case of an erroneous state. The overall user interface and interactions are optimized for efficiency and provide constant feedback, visually or physically, about the system's current state.

- **Navigation:** The user in the immersive VGE should be able to navigate through the digital terrain and change their position and rotation within the given space intuitively. Information about the user's current position is expected to be easy to query and, ideally, visible at all times.

- **Reliability:** Regardless of the imported data set, the system is intended to visualize the data reliably and clearly define the interface and requirements. An erroneous state should not lead to a complete shutdown of the application.

- **Maintainability:** Design choices of the system must consider easy maintainability. For instance, user interaction metaphors in VR are compelled to work with arbitrary input devices in case of changes to the hardware. Lastly, the architecture and integrated libraries got to build a base for future implementations and support specific adaptions in case of requirement changes.

## 3.2 System Overview

This section describes the hardware and software components of the prototype. The framework's design is influenced by several factors: the type of collaboration, hardware, software availability, compatibility, used libraries, and overall requirements discussed in the previous chapter.

The objective is to transfer interactions with the data intuitively in an immersive 3D space. The collaborative setup of the prototype will be asymmetric and co-located. Two users will be able to interact with the terrain data from a 2D display and a VR headset synchronously using the same instance running on a local computer in the same room. For this work, remote collaboration is not needed because the staff exercise takes place in the same physical environment. The goal is to evaluate the potential of immersive VGEs during the staff exercise. Therefore one immersed user is sufficient. Figure 3.3 shows the main components of the hardware and software system and describes the general

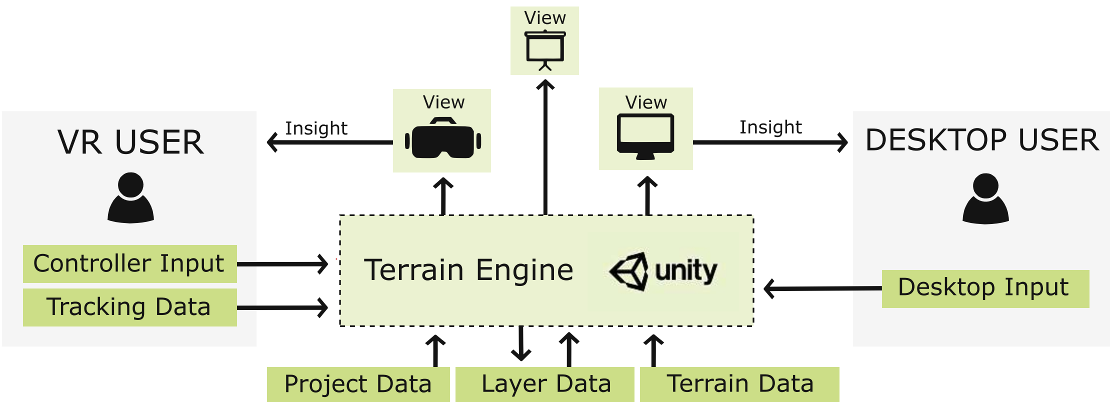


Figure 3.3: Data and information flow of the proposed application.

data and information flow. We use Unity3D, a state-of-the-art rendering engine for the development of the interactive terrain [Tec21]. Unity3D is commonly used for game design; however, it has recently evolved into a leading developmental environment for AR and VR applications. There are numerous VR and AR libraries and plugins available. Examples include: SteamVR [Val21], MRTK [Mic21a], VRTK [Sto21], Vuforia [PTC21], OpenXR [Gro21] and Wikitude [Gmb21]. Basic functionalities of such libraries support rapid prototyping when using Unity3D. Supported programming languages are C# and javascript. The former will be used in this work. The community support and compatibility with off-the-shelf VR hardware make Unity3D an ideal platform to be utilized in the present thesis work.

Unity3D is responsible for rendering different views of the terrain and their annotations in our developed prototype. To this end, Unity3D will target two primary output displays, a state-of-the-art VR headset and a standard 2D display. As seen in Figure 3.3, both users in VR and desktop can extract information from their unique viewpoint. The third view is optional if one additional display is connected to the system for non-interactive purposes (e.g., presentations). In addition to rendering, Unity3D provides management of game logic, a physics engine, components for UI and interaction design, and compiling support for the target operating system, Windows 10.

### 3.2.1 System Devices

The hardware used for the system contains a high-performing desktop computer with state-of-the-art graphics card and processing units, peripherals for the desktop user, and a high-end headset for the user in VR.

Commercially available VR headsets differ in various factors, such as support for hand tracking, eye tracking, controller design, position tracking, stand-alone versus tethered, operating system, display size, and the overall quality of the sensors and software. Every headset fulfills a different requirement based on its quality and sensor configuration. Stand-alone headsets run their operating system and are not dependent on other processing hardware. Headsets in this category are more suitable for mobile applications, lightweight usage, and applications using minimal processing resources. Tethered HMDs use the processing power of another computing device, such as a desktop computer, and, therefore, offer more computing flexibility. Some headsets, such as Oculus Quest 2 [FT21], blend the virtual world with the natural environment, also called Video-See-Through (VST) AR. Another property of HMDs is the method used to calculate the position and orientation of the user, most of the time the head position, in space. There are two types of techniques: inside-out and outside-in tracking. The former tracks the user's head position in the physical environment using several sensors inside the headset by reconstructing their surroundings. On the other hand, headsets using outside-in tracking depend on external sensors calculating the position and orientation of the headset in a given physical space. Currently, outside-in tracking methods are more reliable but require additional space and time to set up the external tracking hardware. We used the HTC Vive Pro [Cor21a] headset for their performance and reliability. This VR headset has a resolution of 1440 x


Figure 3.4: The HTC Vive Pro components. From left to right: the head-mounted display itself, the input devices, also called controllers, the lighthouse sensors used to track the pose of the headset and the controllers, and a wireless adapter. Images taken from the official HTC website [Cor21a]

1600 per eye, a refresh rate of 90 Hz, 110-degree field-of-view, uses outside-in tracking to calculate the pose of the user, and runs on a desktop computer. Figure 3.4 shows the components of the HTC Vive Pro:

- **Headset:** with a dual AMOLED screen, built-in microphones, adjustable Interpupillary Distance (IPD) and integrated headset for audio.
- **Controllers:** represent the position and orientation of the hands of the user and are used as an input device.
- **Base stations:** also called *lighthouse sensors*, track the position of the headset and the controllers.
- **Wireless adapter:** replaces the cable between the headset and the computer and wirelessly transfers data using WiGig, a 60 GHz Wi-Fi connection. This component is optional but enhances the experience because the user does not need to worry about cable management.

The lighthouse sensors create a tracking volume, a physical environment where users can walk around, and their pose is tracked virtually. The sensors sweep the room with an infrared laser scan line, horizontally and vertically respectively, and the receivers on the headset and controllers use the time difference between the laser sweeps and the known spatial relationship of the receiver sensors on the tracked devices to calculate their position and orientation [BSC+18]. The sensors can track users in a physical space up to 5m x 5m and 10m x 10m, depending on the model [Cor21a]. For this work, a minimal tracking space of 2m x 2m is required. The objective is to offer the user in VR a physical space without obstacles to avoid collisions and accidents while interacting with the virtual environment.

## 3.3 User Interface and Interaction Design Concept

As stated in the requirements, usability and intuitive interactions are crucial for this work. This section describes the initial concept for the 2D UI and spatial interactions in 3D for the user in VR.

### 3.3.1 General Guidelines and Considerations

Evaluating user interfaces is a non-trivial task. Various factors influence the effectiveness of human-computer interactions: user state, type of input device, use case, and the shape, color, and function of the Graphical User Interface (GUI). Though, there are general conventions designers can follow. Some design principles, proposed by Nielsen et al. [NM90], are:

- **Simplicity** - Natural and simple interactions with an appropriate learning curve. Avoid clutter or unnecessary sensual elements.
- **Consider the state of the user** - Provide visual communication for the user about the state of the application. Avoid memory overload and include visual, audio or haptic feedback.
- **Consistency** - The design of the interaction metaphors and user interface should be consistent to avoid disorientation and increase overall efficiency
- **Error management** - Indicate faulty states and offer possibilities to exit such cases.

#### User Objectives and Roles

Additionally to the general design guidelines, the design of this work has to consider different user roles and users with various know-how. Table 3.1 provides an overview of the user roles and their goal depending on the phase of the staff exercise.

| | Mission Planning | Briefing/Debriefing | |
|---|---|---|---|
| **User** | Domain Expert | Domain Expert | Commander |
| **Goal** | Annotates the terrain according to the tactical task | Presents the prepared tactical strategies | Analyses the presented strategies, gives feedback |

Table 3.1: Overview of the user roles and their objectives depending on the phase of the staff exercise

During the planning phase, the system users are domain experts, and their goal is to prepare a tactical strategy for a complex task. During briefing and debriefing, their role is passive. The experts present the prepared strategy to the commander. Further, during briefing/debriefing, the commander interacts with the virtual environment, prepared and presented by the domain experts. The objective of the commander is to understand the presented strategies and give feedback for improvement.

### Design for Collaboration

The type of collaboration between users varies depending on the phase of the staff exercise. During planning, domain experts work together towards the group goal, and during briefing/debriefing, the domain experts communicate strategies with the commander and vice versa. An essential objective of this prototype is to optimize the collaboration and support the decision-making process during mission planning.

A crucial part of effective collaboration is effective communication. Communication includes verbal as well as non-verbal cues. In addition to the previously mentioned state-of-the-art design conventions, the design concept of the user interactions follow several principles to assure efficient communication between the user in VR and the user utilizing the desktop:

- **Mirror VR View:** The user on the desktop should be able to see and understand the current task executed by the user in VR.
- **Pointing:** The user in VR should be able to highlight a specific point or object to indicate point-of-interests visually.
- **Selection:** Both users should see if the other user is working on a particular object in real-time. Selected objects are highlighted accordingly.

The following section describes how we used the presented design guidelines to construct a concept of the 2D user interface for the desktop user and the 3D interactions for the user in VR.

## 3.3.2 Design Concept 2D User Interface

Figure 3.5 demonstrates a UI concept derived from the presented design guidelines. The basic components are a menu, to access tools and features to manipulate information on the map, a map view window, preferably full width and height, to visualize the geographical data, a VR view, to show the user on the desktop what the user in VR

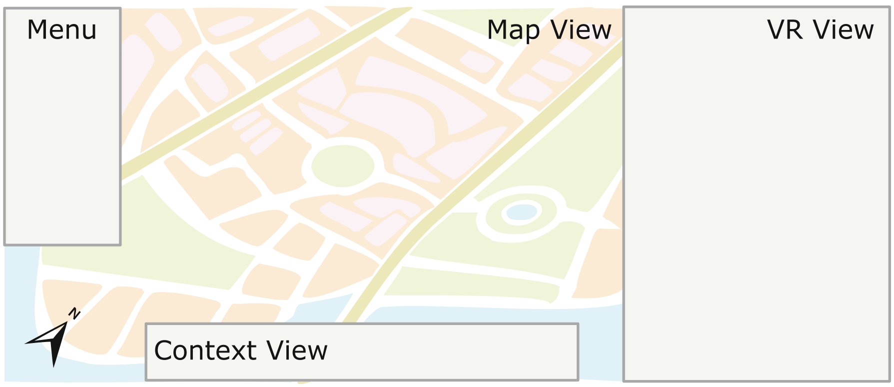


Figure 3.5: UI concept of the desktop interface based on the requirements.

perceives, and a window, the context view, containing queried context and general information related to this work.

Overall, the desktop user interface should maximize the 2D **map view** and its objects. The map view window is in the center and shows the entire or enlarged parts of the map. Therefore, the user will interactively zoom, rotate, and position the map and request geographical coordinates of arbitrary points. Moreover, the user can interact with objects on the map, such as annotations. Temporary information requested by the user should be shown in the **context view**, on the bottom of the screen. Further information essential for the user is geographic directions and the extent of selected objects on the map.

The **VR view** window mirrors the perspective of the user in VR to support cooperative work. The user can interactively change the size of this window to avoid the occlusion of the map itself. This requirement also holds for the **menu** window, seen in Figure 3.5 on the left. The menu provides general functionalities, such as drawing annotations, visibility management of annotation layers and terrain textures, import/export, adjustment of different parameters related to the map, and other tools to support cooperative work.

### 3.3.3 Spatial User Interaction Design

A key objective of a VR system is to present geographical data from unique perspectives and to enable an intuitive way to interact with data. The user interactions and features in VR are expected to be designed to meet these terms.

The desktop user utilizes standard peripherals, such as a 2D computer mouse and keyboard. On the other hand, the input device for the user in VR are two HTC Vive controllers, seen in Figure 3.4, sometimes referred to as *wand*. For this work, two controls on the Vive controller are used: the trigger and the touchpad, shown in Figure 3.6 on the left side. The trigger, a binary input value, is accessed using the index finger and the touchpad, a 2D input vector, using the thumb.


Figure 3.6: **Left:** The two input controls used for spatial interactions, the trigger an the touchpad. **Right:** The touchpad serves as an input for the radial menu in VR.

### Navigation

A large body of related literature proposed several ways to navigate data in an immersive environment. Common locomotion types are already discussed in chapter 2.2.2; however, their implementation heavily depends on the use case of the developed application. The developed prototype for military staff training has to support different terrain dimensions. For instance, in virtual, fine reconstructed cities from drone data, the user can explore the data on a first-person scale. In the first-person scale, the user and the terrain are visualized in a scale ratio of 1:1. However, traveling on a first-person scale on a broad terrain can be cumbersome and counterproductive with an extent of several hundred meters.

The locomotion metaphors for this work will offer support for travelling large and short distances:

- **World-In-Miniature (WIM) [SCP95]:** The user in VR can choose the position on the terrain using a minimized version of the map used to travel large distances.
- **Teleportation:** A target-based locomotion technique commonly used in the first-person scale, used for short distances.
- **Flying:** The position of the user is directed by an input device. Used for traveling on every scale.
- **Walking:** The user can travel within the boundaries of the tracking volume, defined by the HTC Vive Lighthouse tracking sensors. Used for traveling short distances on every scale.

### System Control

A 3D environment gives rise to several possibilities to place graphical elements for system control. Among other things, system control functionalities include selecting the desired locomotion method or placement of annotations. In the case of military staff training, users work rapidly and efficiently. Therefore, this work utilize an accessible and well-known method to implement system control in 3D space: *pie menus*, also called *radial menus* [HCW88]. The touchpad servers serve as an input for the user to navigate and select the desired option of a circular list menu. They are fixed around the touchpad of the VIVE controller, seen on the right image in Figure 3.6. Compared to graphical menus anchored in the 3D space, the advantage of using a pie menu is twofold. First, the user receives haptic feedback by touching the touchpad on the controller, and second, over time, the user might be able to select a menu option from memory, even if the radial menu is out of sight due to the consistency of the menu design. For instance, the center of the touchpad can be reserved for a specific command.

### Manipulation, Selection

Additionally to system control, virtual task planning for military staff exercise involves manipulating and selecting annotations and other objects on the 3D terrain. The metaphor used to manipulate and select objects in the virtual environment is influenced by the distance between the user and the desired object. **Direct manipulation and selection** is used if an object is within reach of the user. The user can manipulate this object using the collision of the object with a controller. The trigger button on the controller is utilized to select a moving object. The position and rotation of the hand of the user can directly adjust the pose of the selected object.

A common interaction technique in VR used for objects out of reach is **ray-casting**. The origin of the ray is on top of the controller, and the direction of the controller controls the direction of the ray. The first object collides with the ray from the ray origin to infinity along the ray direction is selected. If manipulation is allowed, the trigger button on the controller, which is the originator of the ray, is used for further manipulation. During the presentation phase of the staff training, ray-casting can be used to indicate objects of interest.

CHAPTER 4

# Implementation

This chapter discusses details of the implementation of the prototype and highlights emerging challenges and their solutions. The first section plots an overview of the system architecture, including the libraries used and the communication flow. Then, the overall management of the data, such as import and export of geographical data and annotations, is described, followed by two sections illustrating the implementation of functions for the desktop user and the user in VR.

The development followed a participatory design approach. Regular meetings in person and, due to the COVID-19 pandemic, virtual meetings were conducted with consultants from the Austrian military department for geographical operations.

## 4.1 System Architecture

The architecture of the implemented prototype is shown in Figure 4.1. It shows the hierarchy and relationship of the main components. The architecture can be organized into three tiers: presentation, logic, and data. Components in the presentation tier are responsible for the desktop and the spatial user interface, translating information from the logic tier to a human-readable presentation and vice versa. Furthermore, the components in the presentation tier manage the 2D/3D user interface elements. The logic tier encapsulates components coordinating different logical processes of the application. Calculations and decisions based on the input from the presentation and data tier are processed in this layer. The bottom layer, called the data tier, manages the access to data, correspondingly the geographical data, project information, and layer data. The logic tier has writing and reading access to the local file system.

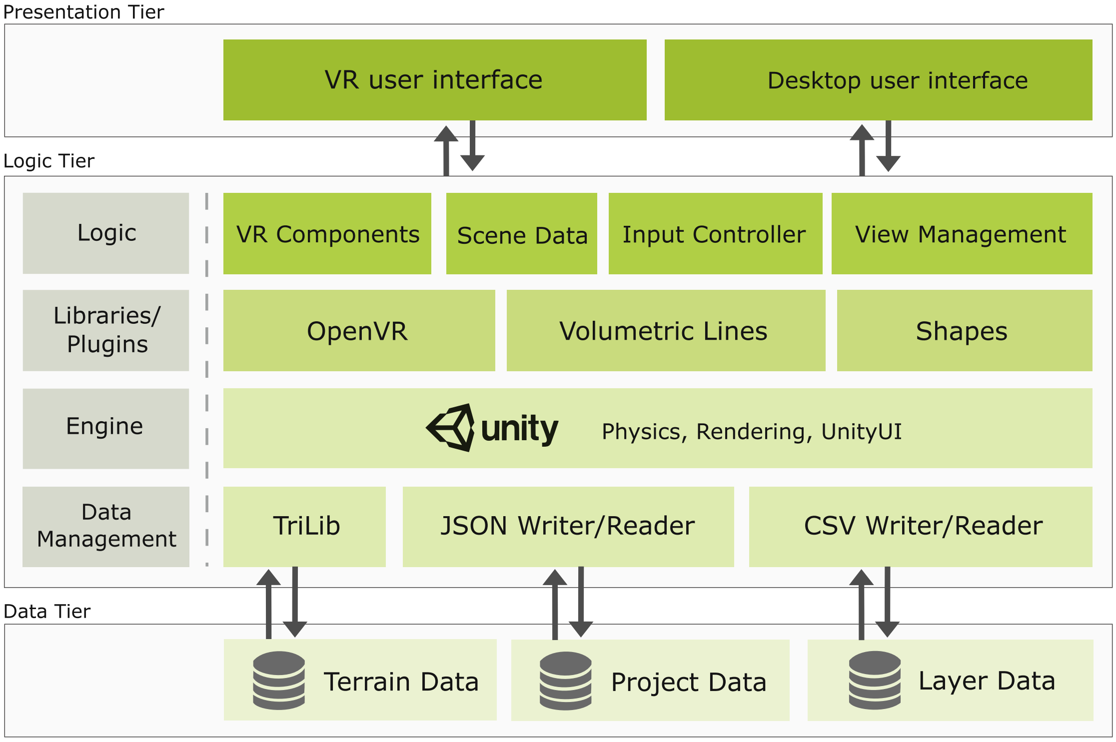


Figure 4.1: The hierarchy and components of the system of the proposed prototype. The architecture components can be grouped into three tiers: presentation, logic, and data tier.

The data tier and data management, including the import and export processes, are explained in detail in section 4.1.3. The software design notes, including object instancing, management of UI elements, and communication between software components, are described in section 4.1.2. In the following, we explain the functions of the rendering engine, Unity3D, and third-party libraries and plugins.

The prototype is built on Unity3D, a proprietary rendering engine and development platform for 2D and 3D applications. The platform offers a visual editor and supports several programming languages. Unity3D carries out the building process for different operating systems with ease [Tec21]. In this work, the following processes are handled by Unity3D: physical properties such as collisions, shading, and management of 2D user interface elements (e.g., sprites). For this prototype, we use `Unity2019.3.1f`.

### 4.1.1 Framework and Libraries

In addition to Unity3D, a collective of other software libraries are utilized for specific rendering tasks. These include: OpenVR [Sof21], the SDK for the HTC Vive, Volumetric Lines [Unt21], a Unity3D plugin for line rendering on the GPU, and Shapes [Hol21], a vector graphics library for Unity3D.

### OpenVR

OpenVR is an interface between VR applications and VR hardware and maintained by Valve Corporation, a well-known publisher for video games. OpenVR offers an Software Development Kit (SDK) and is compatible with numerous VR headsets, such as HTC, Oculus and Microsoft [Sof21]. The Unity3D plugin used for this prototype is called SteamVR, based on the OpenVR SDK and serves as a run-time library [Cor21b]. SteamVR provides the basic building blocks for interactions. This prototype implements the following features with the support of the SteamVR plugin:

- **Virtual 3D model for the controller:** SteamVR provides a virtual 3D model of the physical hardware. The model is interactive and animated and offers a general overview of the controller's status, such as battery health. SteamVR manages the visibility of the virtual controller. In case one of the controllers gets disconnected, SteamVR reacts accordingly.
- **Support for various VR and AR hardware:** SteamVR provides support for state-of-the-art immersive hardware, such as Oculus, Windows Mixed Reality Headsets, and Valve Index, and therefore increase the overall maintainability of the user interaction system in VR.
- **Input mapping:** SteamVR manages the connection between input events and input controls. For instance, the touchpad and the trigger on the HTC Vive Controller. This way, the input logic is separated from the actual input event.
- **Chaperone:** The plugin draws a bounding box for the user, also called *chaperone*, to visualize the edges of the tracking volume. The chaperone is a security feature to ensure that the user stays safely inside the tracking volume.
- **Access to VR properties:** SteamVR offers access to the tracking status, the extents of the tracking volume, and more.

### Shapes

Shapes is a vector graphics library for Unity3D and is used to render primitive shapes, as arcs, circles, lines, and more. One advantage of Shapes is built-in anti-aliasing. Aliasing in VR can lead to undesired effects and is unpleasant to look at. Shapes helps reduce aliasing of primitive shapes and transfers some processing work, such as rendering of primitive geometry, from the Central Processing Unit (CPU) to the Graphics Processing Unit (GPU).

## 4.1.2 Software Design

The design goal of the software architecture is maintainability, reliability, and precise distribution of processing responsibilities.

### System Properties

For this prototype, we use a hybrid software design approach: metaphors of Object Oriented Programming (OOP), as encapsulation and inheritance, single instancing, and global parameters to indicate the current state of the application. The OOP approach helps to keep a high level of control of the communication between objects and repetition of code.

System status properties are stored in singletons, objects reduced to a single instance, or a global parameter file. If a script requires access to, for instance, the extents of the imported terrain, it can approach the terrain singleton, holding all characteristics and run-time objects related to the terrain. Other system properties, such as the status of the data import process, are accessed through a static property file.

### Object Instantiation and Destruction

For developers, Unity3D provides an interactive *What You See Is What You Get (WYSIWYG)* editor, and support for common Integrated Development Environments (IDEs), such as Microsoft Visual Studio, for coding scripts. In the editor, developers can add and adjust the pose of 3D models, cameras, lights, and other objects in a 3D scene, assign scripts, adjust parameters of such scripts and manage the hierarchy and relationship between all objects. Objects in the hierarchy in the Unity3D editor are called *GameObjects* and scripts attached to it have to derive from the Unity3D base class *MonoBehaviour*. Scripts based on MonoBehaviour can define Unity3D events and functions for physics, scene rendering, initialization, input, GUI rendering and decommission. Some functions defined by the MonoBehaviour script are called every frame by Unity3D, while others are only called at the creation or destruction of objects.

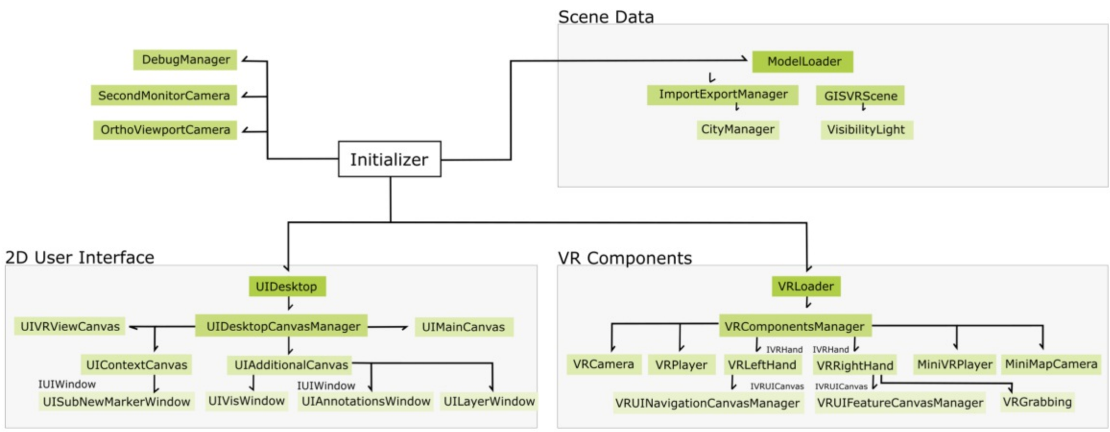


Figure 4.2: A high-level summary of the object generation process. One script generates all other scripts at run-time to reduce manual adjustment in the Unity3D editor.

There are two ways to add objects to a virtual scene: I) by instantiating them via script at run-time, and II) by manually adding them to the hierarchy in the visual editor. Instantiating objects at run-time increases the processing time and, depending on the complexity of the object, might lead to low performance but decreases the manual workload and complexity in the hierarchy in Unity3D's visual editor. We decided to manage all objects in one scene for this prototype and instantiate the majority at run-time. This increases the maintainability of the prototype since changes in the visual editor can not be reliably tracked using version control. Additionally, using code to manage objects offer improved controllability at the start-up. Scripts attached to the objects in the hierarchy of the editor follow a specific execution order, although maintaining the overview of hundreds of objects and their scripts can be challenging.

Figure 4.2 shows a high-level perspective of the object generation process. One script, called *Initializer*, generates all other scripts as objects managing the geographical data, camera management, or desktop user interface at start-up. Our generator metaphor reduces manual adjustments and script management in the Unity3D editor by instantiating objects when needed.

### View Management

In Unity3D, objects can be assigned to a visibility layer. Layer assignments are used to manage the rendering and collision detection of objects. Every camera in the scene renders only specific layers. For instance, the camera used for the desktop map view renders objects assigned to the layers relevant to the desktop user, whereas the VR camera renders objects only related to the user in VR. Despite all objects being present in the same scene in Unity3D, each camera only renders objects designated to the rendering layers assigned to the camera. This saves rendering time and increases performance.

The Unity3D scene, including all objects, is rendered from different viewpoints using six cameras in total. Here is a summary of all cameras and their purposes:

- **VR, right and left eye:** In theory, there are two cameras to render the scene for the user in VR. One from the perspective of the right eye, and the other of the left, slightly shifted to provide stereo vision. We use an optimized stereo rendering technique called *single pass instanced rendering*. It lowers the rendering workload of the CPU by reducing the number of draw calls. The scene graph in Unity3D is iterated once for both eyes, and the rendered view for the left and the right eye is drawn on one texture.
- **Map view:** This camera is used for the main 2D map view for the desktop user, seen in Figure 3.5 and renders the scene orthogonal. The camera parameters, such as zoom factor and positions, are adjusted if the user requests to modify the map section. This camera solely renders objects essential for the desktop user.
- **Magnified view:** This camera amplifies desired sections on the terrain, used to enable precise positioning of scene objects.

- **3rd person view:** This camera is placed behind the user in VR, so the desktop user can observe the behavior and decisions of the immersed user.
- **Mini map camera:** Similar to the desktop map view camera, this camera renders the entire map and presents the current position of the immersed user on the map. The mini map provides the user with an enhanced sense of direction for navigation on the terrain.

#### Scene Lighting

In Unity3D, real-time lighting calculations of virtual objects are computationally intense and usually increase with scene complexity. For this prototype, we do not use dynamic lighting. All shaders used for rendering are based on unlit shaders. Objects are either textured or shaded using a single color.

### 4.1.3 Data Management

A project package for this prototype, imported into the virtual environment in Unity3D, consists of three types of data: *geographical data*, the data for the 3D terrain, *project data*, used to define the details of the geographical data, and *layer data*, representing the annotations created by the user. The following tree depicts the folder structure of a project package:

```
main folder
├── (1) .gisvr
├── (1) .obj
├── (1) .mtl
├── (0 - *) .png .jpg .jpeg
├── textures
│   └── (0 - *) .png .jpg .jpeg
└── obj
    ├── markers-png
    │   └── (0 - *) .png .jpg .jpeg
    ├── screenshots
    │   └── (0 - *) .png .jpg .jpeg
    └── (0 - *) .csv
```

The first three files, the project data (`.gisvr`), the Wavefront file (`.obj`) and its material data (`.mtl`) are required. All other files and folders are optional. The main textures of the terrain data are saved in the main folder. Additional textures are placed in a subfolder called *textures*. The subfolder *obj* holds additional data such as table files for terrain annotations (`.csv`), a folder named *screenshots* for image snapshots saved by the users and a folder *markers-png* for the import of marker symbols.

### Geographical Data

The 3D terrain data is encoded as an OBJ file. The OBJ file format contains vertex data, faces, texture coordinates, vertex normals, and more. The 3D data is interpreted and imported at run-time using a Unity3D plugin called *TriLib 2.0* [Rei21]. One advantage of the TriLib is its threading approach. The geographical data is read parallel to the main thread and therefore does not block other running processes such as the main Unity3D scene rendering thread.

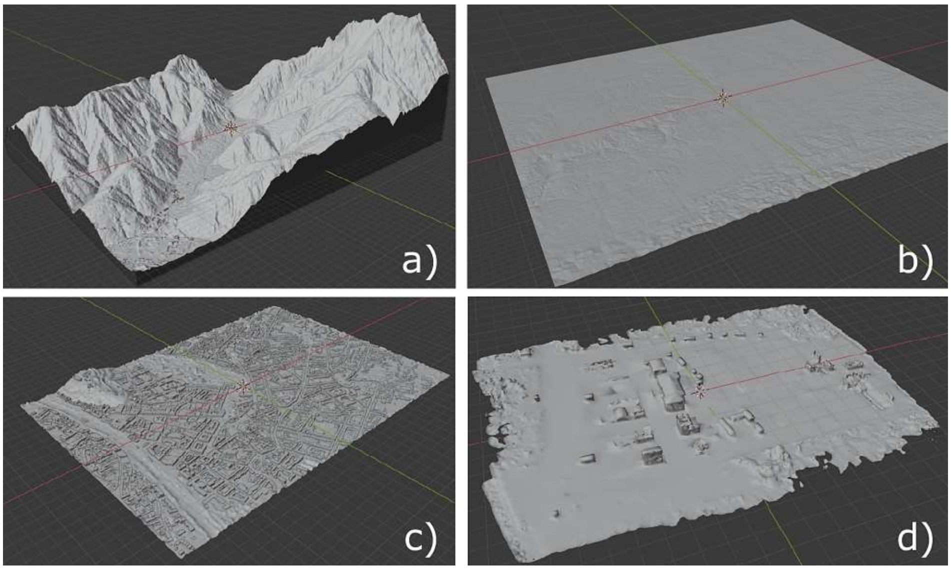


Figure 4.3: The properties of the used terrain data vary strongly. **a)** A fine reconstruction of a mountain range, **b)** a wide map and coarse definitions of riverbeds and hills, **c)** terrain derived from DEM data, consisting of a base map and building data and **d)** a reconstruction of military exercise grounds, based on drone images.

Selected terrain in Figure 4.3 shows the diversity of terrain properties. The terrain data obtained from the military department of geography, IMG, vary from a fine triangle mesh, reconstructed using aerial images, to urban data, consisting of flat ground and detailed buildings on top of it. The prototype has to understand the properties of the geographical structures, such as extents, textures, benchmark, and alignment. Therefore, the terrain data has to be pre-processed manually in Blender, an open-source 3D computer graphics software [Fou21], to provide compatibility with the coordinate system in Unity3D. Specific properties are required the ensure an accurate import of the terrain data from Blender into Unity3D:

- **Ground data** - An essential part of the terrain data is the mesh of the ground itself. Additional mesh, such as buildings or other objects, should be separated from the ground mesh.
- **Geometric center at (0,0,0)** - The geometric gravity of the 3D model is the center of the scene in Blender, at (x0, y0, z0).
- **North alignment** - The 3D mesh is located in the x, y world plane, and the +z axis pointing upwards. North is in the direction of the +y axis in Blender. This requirement helps align the terrain correctly in Unity3D. Blender uses a right-handed coordinate system, and Unity3D's system is left-handed.
- **Known benchmark** - The benchmark of the 3D model in Blender should be specified in the project file (`.gisvr`) correctly to ensure exact measurements and coordinates in Unity3D (for instance, 1 unit in Blender = 1 m and the terrain is scaled accordingly. then the respective benchmark is 1:1).
- **Optional collision mesh** - In addition to the terrain mesh, a collision mesh can be defined in Blender. If no collision mesh is defined, Unity3D duplicates the triangle mesh of the terrain and uses that one to calculate collisions. This can lead to intense collision computations, depending on the number of vertices. We recommend generating the collision mesh in Blender, using an optimized version of the terrain mesh.
- **Optimized mesh** - Every triangle mesh should be optimized and the number of triangles reduced to a minimum without altering the mesh's topology.
- **Minimal extents** - Large-scale terrain models (> 50km in one direction) are not ideal due to floating-point issues in Unity3D and Blender.
- **Texture Mapping** - The textures mapped to the geometry in Blender are exported to Unity3D as is.

The requirements on the terrain data described above support the correct presentation of the terrain data in Unity3D after the import process. The project file (`.gisvr`) holds further information necessary to ensure a geographically correct import into Unity3D, such as proper scale and placement in Unity3D's coordinate system. Listing 1 shows an example project file in JavaScript Object Notation (JSON) notation. We use Unity3D's parser to read and write JSON files.

```
{
    "vrwalking_allowed" : "true"
    "center" : [431194.02,5009344.97,100.01],
    "scale_benchmark" : [1,1],
    "EPSG" : 32633,
    "cities" : [
        {
            "id" : 0
            "name" : "Location0"
            "UTM" : [431192.05, 5009336.65]
        },
        {
            "id" : 1
            "name" : "Location1"
            "UTM" : [431213.92, 5009333.86]
        }
    ]
}
```

Listing 1: Structure of the project data file based on JSON. The project file accompanies the geographical data.

In listing 1, the first parameter `vrwalking_allowed` sets the level of accessibility to the terrain for the immersed user. If the value is `true`, the user in VR is allowed to navigate the terrain on a first-person scale. The following parameters in the project file are used for correct geographical interpretation of the terrain data: `center`, defines the UTM coordinate of the origin - (x0, y0, z0) - in Blender, `scale_benchmark` specifies the scale benchmark of the terrain data and `EPSG` is a key for the referenced geographical coordinate system. The last parameter, `cities`, defines a data array used to import landmarks on top of the terrain into the immersive scene. Well-known landmarks are used to support familiarity with the map. In particular, a user in VR can estimate their position on the map based on known landmarks in the scene. Every landmark consists of an id, label, and an UTM coordinate.

### Layer Data

An important use case derived from the field study and application requirements is placing 2D and 3D annotations on the terrain. The desktop user and the immersed user can place markers, point-of-interest, and lines according to their tactical strategy. The annotations are organized in layers, and every layer is saved in a single table file `.csv` in the local file system. The layer files represent the output of the planning phase of the military staff exercise. The layer data is imported at the start of the application. Both users can export the layer data at any time.

Listing 2 shows an example table file of one annotation layer. All elements in the table

```
TYPE;ID;LABEL;SYMBOL;EPSG;WKT;COLOR;SIZE
P;P_E324961N5260619;leer;-;32633;POINT(325527.395 5261362.223 193);
rgb(0,0,255);
L;L_12291Q5;leer;-;32633;LINESTRING(325122.56 5260430.54 1901,...,
325494.12 5261495.32 1902);rgb(0,255,0);
F;A_11295HS;leer;-;32633;POLYGON(324556.9 5260785.47 1604,...,
323464.39 5261611.78 1057);rgb(0,255,0);
S;S_E547878N5362368;1.Kp,33T VN 93819 65292,h:803m;AufklZg.png;32633;
POINT(547878.11 5362368.38 199.84);
L;V_457ZMN;-;-;32633;LINESTRING(572697.4 5369282.75 6903.02,...,
571278.67 5370069.81 7467.48);rgb(0,255,0);0.1
```

Listing 2: Structure of a layer data file defining points (P), lines (L), areas (F) and markers (S)

belong to this layer. Every annotation element has a unique identifier, label, an optional image or symbol, EPSG code, encoding for the geometry, and color. The geometry is encoded using a markup language for vector geometry, called *Well-known Text (WKT)*. The WKT representation offers a standard format for geometries and allows officers to reuse the layer data for other GIS systems. WKT coordinates are UTM positions, indicating a unique location and height.

- **Point-of-interest, (P)** - Mainly defined by its UTM coordinate, id and color. Represented by a 3D sphere
- **Line or path, (L)** - Several points define a line. A 3D line has an additional parameter, defining the width of the line
- **Area-of-interest, (F)** - Similar to a line, but the last point is connected to the first
- **Marker, type (S)** - A marker is defined by a point and a symbol, saved as an image

### Data Import

Figure 4.4 shows the details of the data import process. At the start of the application, the user on the desktop is asked to import a project by selecting a project folder, a folder containing the project file (`.gisvr`). After selection, the terrain data, including immediate textures, are imported using the Unity3D plugin TriLib. After a successful import, additional terrain textures and potential layer data are parsed and processed. Only after the import of all data files, the UI elements for the desktop are loaded at run-time. During the main state, the prototype already processed all project data and is waiting for input events and commands from the users. The VR headset is not loaded automatically, but the desktop user can enable and start the VR mode any time during

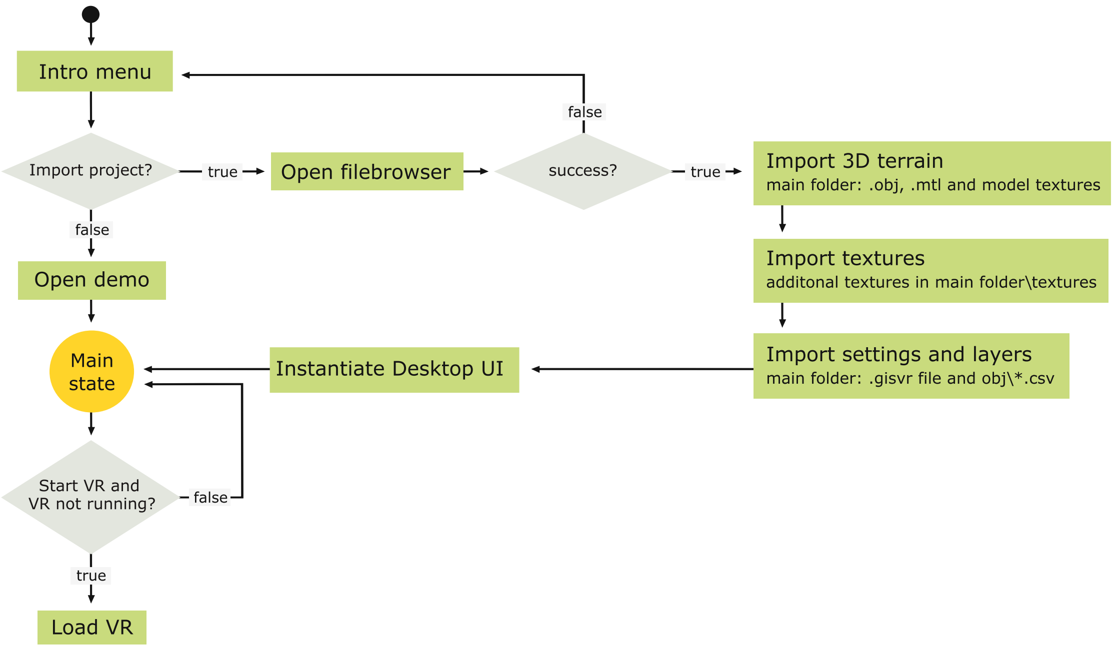


Figure 4.4: Flow chart of the processes at start-up.

the main state. As soon as the VR hardware is loaded, a second user can access the system in VR.

### Terrain Data Registration

The gravity of the terrain data in Unity3D is the origin of the coordinate system (x0, y0, z0). The main map projection used for this prototype is UTM [Lan98]. A map projection is a method to present a curved surface on a flat plane. The UTM assumes the shape of the earth as a spherical model, and for this prototype, the referenced geographic datum is WGS84 [Kum88]. A geographic datum defines important properties of the geometric model, such as the radius and axis of the surface of the reference sphere and the position of the surface relative to the earth's center. Using UTM, the earth is divided into 60 zones, with a longitude width of 6°. A location on the map is defined by its zone, along with two coordinates for easting and northing. The distance between each coordinate is one meter and, therefore, very convenient in metric coordinate systems as in Unity3D. The height information of the center point, imported from the project JSON file, is used as a reference to calculate the metric height of the other points on the terrain.

One challenge was to manage different extents of imported terrain data correctly. As shown in Figure 4.3, the extent of the physical space of the terrain data differs by several orders of magnitude. For instance, the terrain shown in Figure 4.3b, has a dimensionality of $76600m \times 69102m$, whereas the area presented in Figure 4.3d, is of $204m \times 157m$ size. Coordinates in Unity3D are defined by floating-point numbers, and precise placement decrease if the distance of a point to the world's origin increase. The position and appearance of objects in space are unstable the farther away from the origin, an effect called *Spatial Jittering* [Tho05]. Our tests showed that such unpleasant effects occur starting from $1000m$ distance to the origin of the world space. This is especially problematic if VR headsets are involved. Due to the immersive point of view, the visual effect of jittering objects, such as user interface elements or virtual hands, could be very unpleasant. We solved this problem by introducing a fixed scale box when importing terrain data. The extents of the fixed scale box are $1000m \times 1000m \times 1000m$. If the largest dimension of the 3D terrain exceeds 1000m, the terrain is scaled to fit the extents of the fixed scale box. Should the largest dimension be smaller than 1000m, the terrain is up-scaled to fit the scale box.

## 4.2 Desktop Interactions

This section illustrates the implementation of the desktop UI and its provided scope of operations for the desktop user. We further explain the objectives and purpose of implemented UI design elements.

### 4.2.1 Desktop User Interface

We used the Unity3D packages *UnityUI* and *TextMesh Pro* to create the desktop UI. The former is a core package used to manipulate UI elements, and the latter is used to render text components efficiently using custom shaders. The GUI elements are arranged in the Unity3D visual editor and saved as a Unity3D *Prefab*. A Prefab represents a hierarchy of objects configured to the developer's needs and stored as a separate file. Prefabs are then instantiated at run-time. If there are several instances, changes to the original prefab are synced to all instances. In this prototype, C# Scripts responsible for interface elements register user interface events, such as clicking or hovering events and assign colors to buttons and icons at run-time.

Figure 4.5 shows the 2D UI elements, their arrangement, and a preview of how we implemented our design concept. The top row presents the abstract design (left) and the implemented interface (right) if VR is not enabled. The desktop user can enable VR access to the application utilizing the *Start VR*, on the top right (Figure 4.5e). The bottom row shows the arrangement of the UI when VR is activated. The color block in figure 4.5 on the bottom right presents the color scheme used for the UI. We focused on main neutral colors, a greyscale range, and shades of blue for accents. The red and green tones are used for specific UI elements, such as confirmation and cancel buttons. The font type used for all text elements is *LiberationSans SDF*, a sans-serif font type provided by TextMesh Pro. The overall design and implementation of the tool menu (Figure 4.5b), system menu (Figure 4.5d), the context view (Figure 4.5c), and the different realization of the map view (Figure 4.5a and Figure 4.5f) are explained in the following sections.


Figure 4.5: The arrangement and implementation of the UI elements. **Left Column:** An abstract overview of the main UI elements. **Right Column:** The implementation of the UI design. The top row presents the UI if VR is not enabled, and the bottom row depicts the arrangement of the elements after activation of the VR mode

### 4.2.2 Views

As discussed in section 3.3, the desktop user should be able to observe the operations of the immersed user intuitively. The desktop interface offers several views for the desktop user. A centered, interactive view of the map (Figure 4.5a) and a VR view (Figure 4.5f). The objective of the full-width map view is to offer a familiar task space for terrain annotations, similar to the paper maps used during the military staff exercise, and a base for visibility analysis and distance measurements. The desktop user can use the *mouse wheel* to enlarge the map, *right-click* to rotate the map around its up-vector, and *right-click + shift* simultaneously to drag the map. The user can reset all manipulations to the position and rotation of the map view using the north arrow on the bottom left of the desktop UI. For direct manipulation of terrain elements, the desktop user can select objects on the terrain, for instance, points, lines, areas, or markers, using a *left-click*. Information about the selected objects, such as the UTM coordinates or dimensions, are displayed in the context view (Figure 4.5c) at the bottom of the screen. Selected objects can be deleted using the *delete button* on the connected keyboard.

The VR view is activated by the desktop user using the button on the top right of the screen (Figure 4.5e). The width of the VR view window is adjustable to avoid map occlusion. The desktop user can change the size by dragging the left handle of the

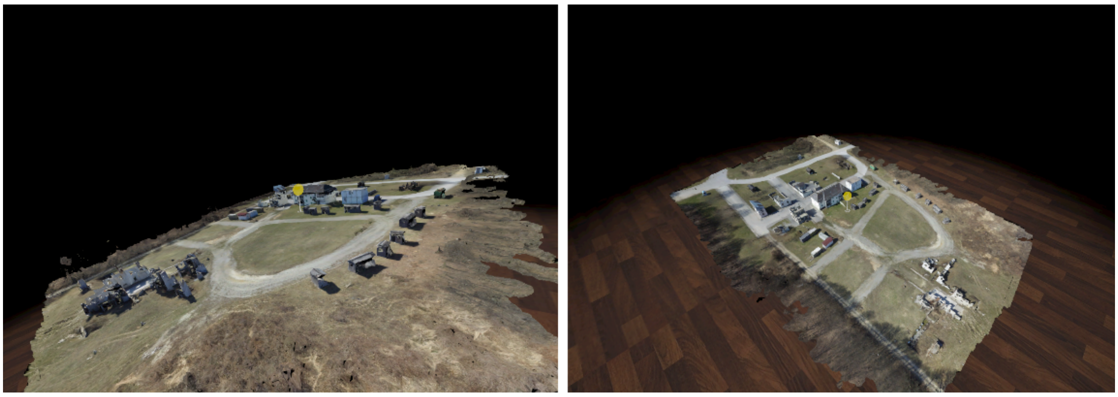

Figure 4.6: The desktop user can choose between two different perspectives to observe the actions of the user in VR. **Left:** A first-person point of view and **Right:** a third-person point of view

window. At the top left corner of the VR view window, (Figure 4.5g), the desktop user can change the viewing perspective of the VR window. One perspective mirrors the output of the right eye of the VR camera, and the other one renders the tracking volume of the immersed user to mimic a third-person's point of view. Some example perspectives are shown in Figure 4.6. The primary purpose of the VR view is to provide visual cues to support user collaboration by providing the gaze direction of the immersed user. During the presentation phase of the military staff training, the prototype aims to support the communication between the staffs and the commander. The commander is interacting with the scene in VR, and one officer is using the desktop interface to understand the actions of the commander. Nevertheless, the other officers in training should also be able to follow the actions made by the desktop and the user in VR. We decided to include an additional view for another display device which can be either a projector or a second monitor. This presentation view mirrors the desktop map view, seen in (Figure 4.5b), showing the terrain and annotations elements. The system menu (Figure 4.5d) holds elements to control the state of the application. The desktop user can take a snapshot of the 2D map view, export all annotations and layer data, and terminate the application.

### 4.2.3 Tool Menu

The tool menu (Figure 4.5b) is equipped with several functionalities for the desktop user for tactical planning and annotation management: An interactive window for annotation layer management, 2D drawing and marker placement, and management of additional terrain texture layers.

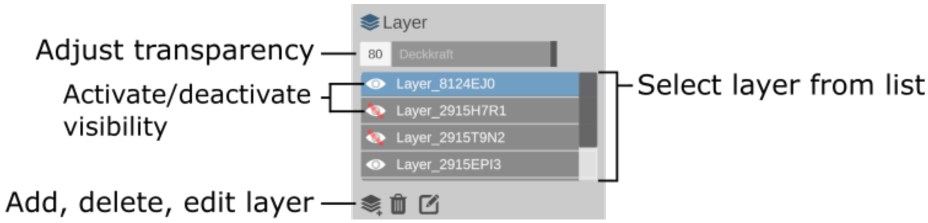


Figure 4.7: The desktop user can adjust the appearance of the annotation layers via the layer management window in the tools menu.

### Layer Management

The user on the desktop can create annotation layers using the appropriate command in the layer management window. An annotation layer is an abstract object arranging annotation elements, such as points, lines, and areas, into groups. Every layer group is exported to a separate layer file and can be imported to another instance of the same project. The layer window, as seen in Figure 4.7, provides the user commands to create, delete and edit a selected annotation layer. Unity3D implements simple commands to create, delete, and adjust objects in the virtual scene. If the user deactivates a layer, the entire hierarchy of that layer is deactivated. All child objects, including the annotation elements belonging to that layer, are then invisible to every camera. The layer window has a slider and a text input to adjust the transparency of a selected layer. For coarse adjustments, the user can use the slider and for an exact definition of the layer transparency, the user can input a value from 0 to 100 into the text input box. A value of 100, presents $alpha = 1$, the highest visibility. The transparency of all objects assigned to a layer is synchronized with transparency of the layer itself.


Figure 4.8: The desktop user can adjust the appearance of the imported terrain texture layers via the texture management window in the tools menu.

### Texture Management

The texture management window, see in Figure 4.8, has a similar structure as the layer management window. The desktop user can adjust the visibility of a texture by changing the alpha value in the slider or the text input box. Terrain textures can be deactivated and activated entirely with a left-click on the texture icon seen in Figure 4.8. A maximum of five textures, including the textures imported with the 3D terrain model, are allowed.

All textures are combined on the GPU in the fragment shader using the weighted average of the color values of overlaid textures. The color of a fragment element is calculated using the following equation for alpha composition:

$$\frac{\sum_{i=1}^{n} on_i * a_i * (R_i, G_i, B_i)}{\sum_{i=1}^{n} on_i * a_i} = (R, G, B) \tag{4.1}$$

Where $n$, $0 <= n <= 5$, presents the total number of imported textures, $i$ is the index of the current texture, and $on_i$, a boolean value, indicating if texture $i$ is visible or not. The user sets the blending value $a$ per texture $i$ using the slider or text input box in the texture window. If the denominator, $\sum_{i=1}^{n} on_i * a_i$, is $< 1$, for instance, in the case no texture is enabled, we use a Material Capture (MatCap) texture to present the structure of the terrain geometry and blend it with the remaining visible textures. MatCap shading is a low-performance method to imitate lighting without actual lighting calculations [SMGG01]. Figure 4.9 presents the output of a standard MatCap shading technique. Figure 4.9a shows a textured terrain. Figure 4.9b depicts the reference texture encoding the lighting setup applied to the terrain. In the vertex shader, we derive the normal vector for every vertex and map the xz components between $[0.2, 0.8]$. The resulting vector is used to query the color for the current fragment from the UV space of the reference texture. Remapping the values avoids the edges of the MatCap reference texture outside the edges of the sphere in Figure 4.9b. The MatCap terrain shading facilitates geometry structure without visual distractions presented in the regular landscape texture. As seen in Figure 4.9a, small ground structures are not visible in the textured terrain on the left but noticeable using the MatCap shader, in Figure 4.9c.

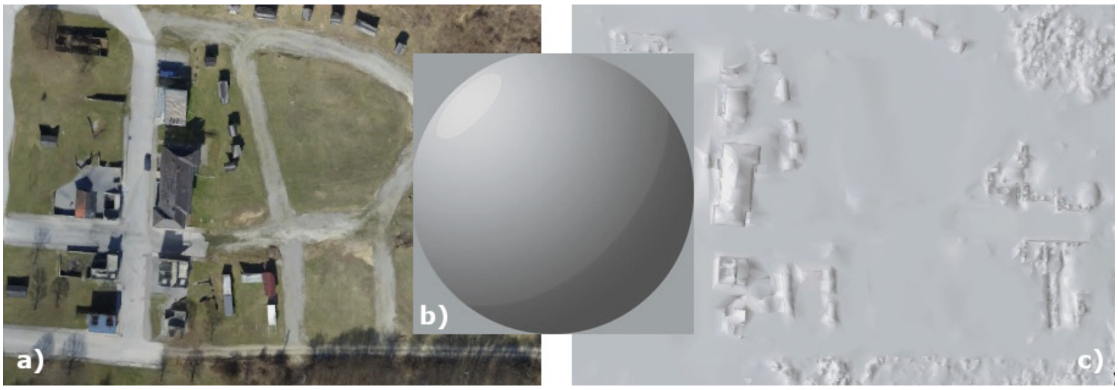


Figure 4.9: The Material Capture (MatCap) shader is used to enhance terrain structure. **a)** The textured terrain, **b)** the reference texture encoding the lighting setup and **c)** the same terrain data rendered with the MatCap shader.

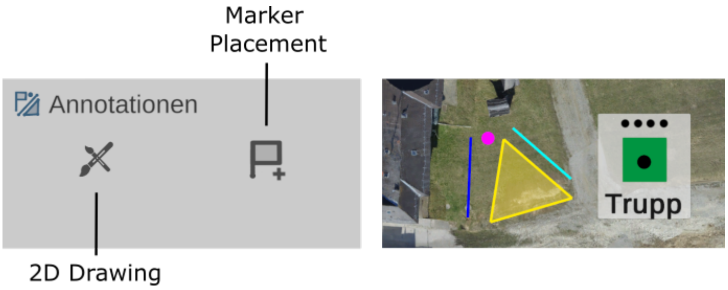


Figure 4.10: The annotation window provides tools for drawing on the terrain and placing military markers. **Left:** The annotation window and **Right:** annotation elements on the terrain, such as points, lines, areas and markers, from the perspective of the desktop user.

### 2D Drawing

During the planning process and decision-making of the staff training, the officers and the commander use maps to annotate points of interest and more. While our prototype focuses on interactions in VR, a user should be able to make quick adjustments to the terrain annotations without a connected VR headset. Using the annotation window in the tools menu, seen in Figure 4.10, the desktop user can set points, lines, areas, and markers on the terrain. An annotation element placed by the user is assigned to the currently active layer selected in the layer window. To enhance collaboration between the user in VR and the desktop user, annotations are visible for both users.

If the desktop user selects the drawing symbol, the drawing mode for points, polylines, convex and concave polygons is activated. Polygons with intersecting lines are not supported. During the drawing mode, the desktop user can choose the desired color for the prospective annotation element. In drawing mode, the UTM coordinate of the current location of the desktop cursor is shown in the context view on the desktop. Using consecutive *left-clicks* on the terrain, the user can draw a line. If the last placed point position concurs with the first placed point, a polygon is created. A *double right-click* places a point. During drawing mode, the user can press the *delete* button on the keyboard to stop the drawing mode without placing any elements.

**Points:** For the calculation of a point location, we use a standard ray intersection test. A ray is sent from the screen position of the mouse along the global negative y-axis, the viewing vector of the map view camera. The intersection point with the terrain is the location for the desired point of interest. A point is represented by a simple 3D sphere.

**Polylines:** A polyline consists of at least two points, and consecutive points are connected by a line, rendered using the Unity3D library *Shapes*. The library draws lines as billboards facing the camera utilizing the computing power of the GPU.

**Polygons:** Generating the geometry for convex polygons is straightforward since we assume that the desktop user draws the points sorted along the hull of the polygon. Should the user draw a concave polygon, we deploy a standard ear-clipping algorithm [EET93] to

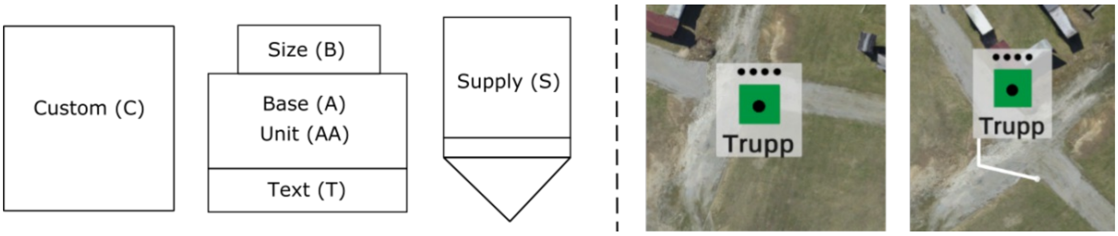


Figure 4.11: The left three images present the structure of the three marker types used in this prototype: custom, composited, and supply marker. The two images on the right show an example of a composited marker. The desktop user can move the marker to avoid occlusions. The exact position of the marker is then indicated by a line from the marker position to the edge of the marker.

calculate the mesh geometry of the polygon. No self-intersections or holes are supported.

**Markers:** The second symbol in the annotation window, Figure 4.10, is used to place military markers on the terrain. We use symbols based on the NATO standard called NATO Joint Military Symbology [NAT21]. The standard provides symbols for military operations and units on land, air, space, or sea. In the case of the Austrian military, the arrangement of the marker components is precisely defined, as shown in the three images on the left in Figure 4.11. Based on the conducted field study and initial interviews with our partners at the military geographical operations team, this prototype supports three different marker types:

- **Composited Marker:** This marker type is a composition of several image elements:
  - **Base element (A):** A color coded affiliation for the subject of the current marker. For instance, a green square is a neutral subject.
  - **Unit element (AA):** Elements of this type are stackable and modify the meaning of the base element. For instance, a green square as a base and with a filled black dot in the middle for a unit element is a symbol for a neutral artillery. If there is an additional black triangle below the dot, it is a neutral mountain artillery.
  - **Size element (B):** Indicates the military size of the current marker. For instance, one black dot defines a squad, two dots a section and three dots a troop.
  - **Text (T):** Optional name of the subject, such as commander or other military units.
- **Supply (S):** A NATO standard military marker type with a special design for supply elements and consists of one image element.
- **Custom (C):** This marker type is reserved for imported markers symbols and consists of one image element.

As soon as the desktop user activates the marker placement, the user has to choose a location on the terrain using a *left-click*. A window will appear, where the user can choose the desired marker type and specifics about the marker, as symbols and text. The unit element of the composited marker type consists of stackable symbols. If the marker placement is finished and the project's current state is saved, all image elements of the composited marker are merged into one image file and saved to the local file system for later use. If a marker is imported from a layer data file, a custom marker is created. The two images on the right in Figure 4.11 show an example of a placed marker. The desktop user can move a placed marker using the desktop mouse to avoid the occlusion of information on the terrain. If the user moved the marker from its original location, a white line is drawn from the left corner of the marker to the original marker location, as seen in the most right image in Figure 4.11.

## 4.3 Spatial Interactions

Compared to user interactions on a 2D display, the design complexity of intuitive 3D user interactions, also called *spatial interactions* is higher due to input devices with higher Degrees-of-Freedom (DoF). Both HTC Vive controllers are tracked in a 3D space and have several input controls, such as buttons and other knobs. In some cases, the spatial relationship between the controllers serves as an input parameter. With this design choice in mind, we had to create suitable concepts for navigation, system control, and object manipulation in VR.

### 4.3.1 Spatial Controls

As discussed in the design chapter, we decided to use the most intuitive input controls on the VIVE controller, the trigger, and the touchpad, seen in Figure 3.6. The former returns a boolean and a float value, representing the selection and magnitude of the trigger button, respectively, and the latter returns a boolean value and a 2D vector, selection, and position of the thumb on a 2D grid. The range of the 2D grid is $x = [-1, 1]$ and $y = [-1, 1]$ with $[0, 0]$ in the center of the touchpad. The touchpad serves as an input element for the radial menu in VR, the radial menu, shown in Figure 4.12. The radial menu is a flat graphical element in a disc shape, consisting of several segments, and every segment presents a command. A segment of a radial menu either triggers a direct command, e.g., taking a screenshot, serves as a toggle button, e.g., activating or deactivating the VR pointer, or lead to a sub radial menu. The menu floats above the touchpad, and the user can select a segment on the radial menu using the touchpad. The current position of the thumb on the 2D vector field of the touchpad determines the selected segment, and pressing the touchpad with the thumb confirms the selection.

The interactions in VR are grouped by their purpose. Commands related to changing perspectives or navigation are placed on one controller, the *navigation controller*, while functions related to state changes of objects or the overall system are placed on the other controller, called *system controller*. Every VR controller holds several user interface

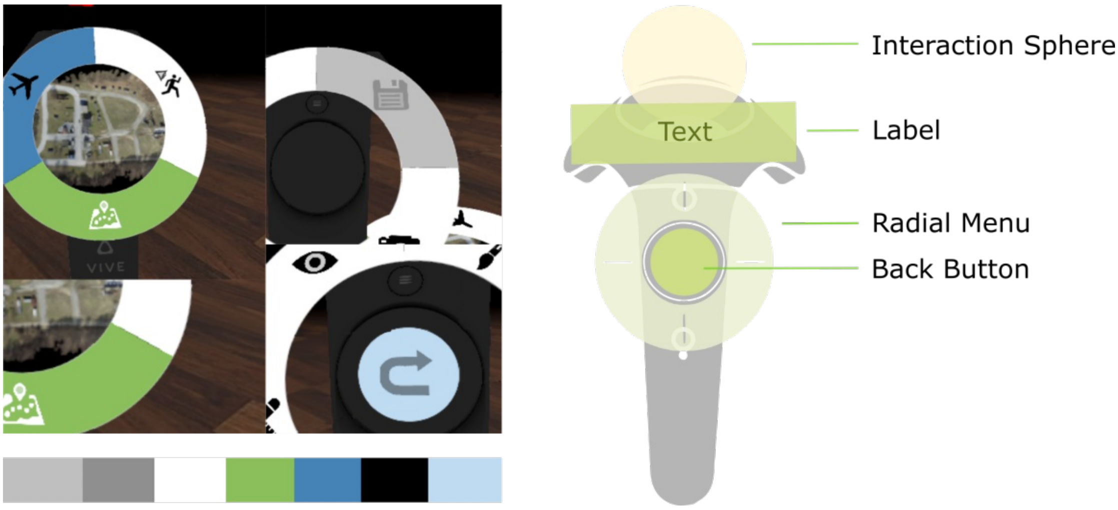


Figure 4.12: The color design and spatial arrangement of the user interface elements in VR. **Left:** The color design of the radial menu. **Right:** The arrangement of the user interface elements.

elements: a radial menu, a back button, a label, and an interaction sphere. The spatial arrangement of the UI elements is shown in Figure 4.12. The interaction sphere is used for spatial selections, for instance, deletion of objects in the scene, and the label shows a high-resolution text element indicating the currently selected segment on the radial menu. The color bar on the bottom left shows the colors used for the VR interface elements. The green shade suggests active commands, dark blue is used for selection, and the grey values indicate disabled user commands.

### Navigation Controller

One challenge was to provide suitable methods for navigation in VR for various types of terrain data. For instance, coarsely reconstructed environments do not provide notable advantages when explored from a first-person perspective, and traveling a terrain with an extent of several kilometers using teleportation is not feasible. This prototype provides

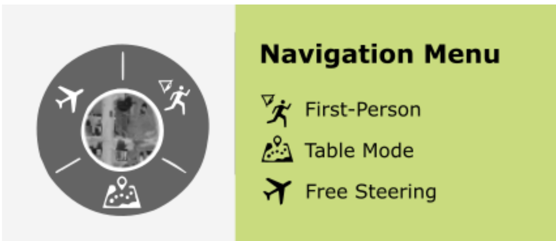


Figure 4.13: The structure of the radial menu designed to help the user in VR to get around the virtual terrain.

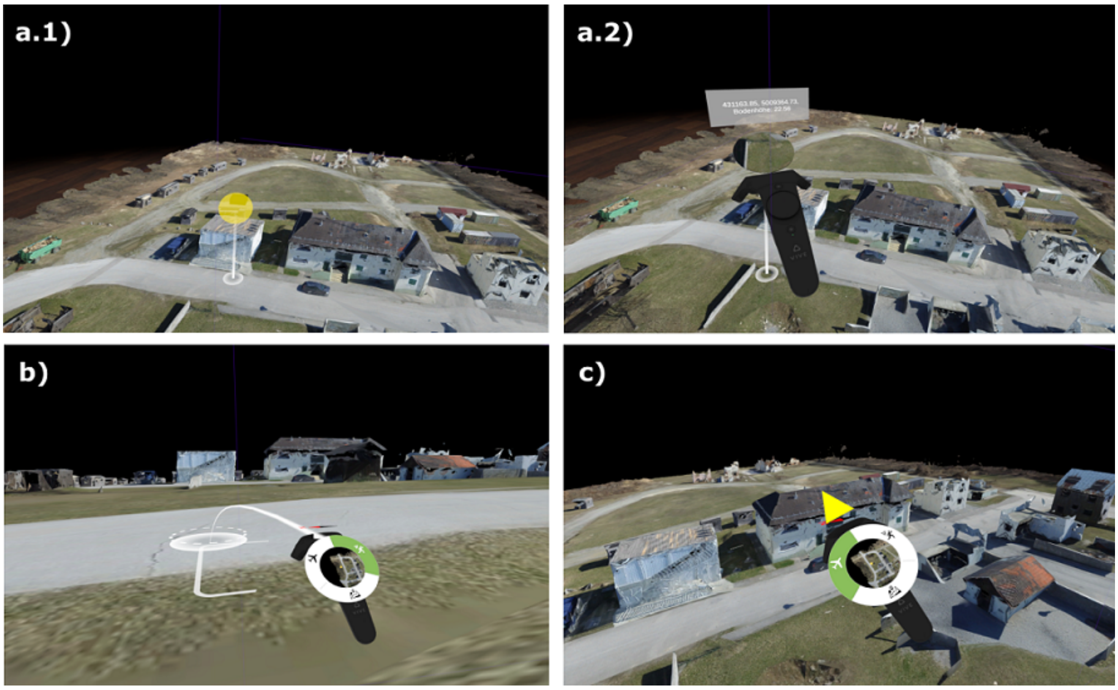


Figure 4.14: The prototype provides three different navigation methods for the immersed user. **a)** The table mode. Image **a.1)** shows the avatar of the user on the terrain, and image **a.2)** shows the magnified view and the metadata, such as terrain coordinates, while the user holds the avatar. Image **b)** presents the first-person navigation using teleportation and image **c)** free steering navigation using the trigger.

several types of navigation for the immersed user to overcome this limitation.

We decided to implement three different methods to navigate the geographical environment in VR to provide meaningful locomotion for various data types, resolutions, and travel distances: a table mode, first-person navigation, and free steering. The user in VR can select the desired navigation mode on the radial menu. The options on the navigation radial menu are shown in Figure 4.13. In the center of the radial menu, seen in Figure 4.14b, a miniature map indicates the position of the immersed user on the terrain. Above the controller, a compass offers additional guidance and improved orientation when changing the navigation mode.

**Table Mode**: World-In-Miniature (WIM) locomotion is a standard technique for large-scale traveling and offers insights on the terrain structure, similar to a workbench view. The scale of the user in VR is adjusted to the scale of the physical tracking volume, so the user in VR perceives the geographical data as a miniature model, seen in Figure 4.14a. The design of the WIM mode resembles a table with an elevated map on top of it. The immersed user can walk in the physical space safely to explore the entire terrain. Similar to the system proposed by Elvezio et al. [ESFT17], the user in VR can choose a position on the terrain using an avatar. The avatar can be moved using the trigger on the HTC Vive controller. After pick-up, the user sees a magnified version of the potential location along with the UTM coordinate for precise positioning, depict in Figure 4.14a.2. The magnified view is generated using a separate camera positioned below the avatar to generated a magnified view of the desired location. Next time the user changes to first-person or free steering mode, the immersed user will be teleported to the avatar's location on the table.

**First-Person View**: The first-person navigation mode, shown in Figure 4.14b, offers a terrain walk-through from an egocentric perspective. The ratio between the terrain and the user is 1:1. The user can choose the position on the ground via a ray, also called teleportation. If this navigation mode is active, the user can activate a curved ray by pressing the trigger of the navigation controller. As long as the trigger is pressed, the user sees a preview of the new location, including the boundary of the tracking volume. Within the tracking volume, the user can move by walking. The ray and terrain intersection is selected as a new user position when the trigger is released. First-person is advantageous for short-distance traveling and exploration of fine reconstructed terrain, as urban environments. The user can get familiar with the terrain on a scale analogous to reality.

**Free-Steering Mode**: The user-terrain ratio in the free-steering mode is the same as in first-person. The difference is that the user can pilot through the terrain without gravity, similar to flying a helicopter. As soon as this navigation mode is activated, the user can choose the desired direction using the pointer on the tip of the controller, seen in Figure 4.14c, and the trigger button. The steering speed increases over time the trigger is pressed.

### System Controller

On the system controller, the user in VR can access controls for system functions and terrain tools, such as annotations and visibility analysis. Figure 4.15 shows a summary of the menu hierarchy of the system controller. The center of the trackpad on the system controller is reserved for the back command, which leads the user to the previous level in the menu hierarchy.

The first level of the system radial menu holds general commands, as saving, taking a screenshot, and VR pointer activation. The first one exports the current state of the system, including annotations and layers. The screenshot function activates a camera preview in VR. With the preview, the user in VR can take a snapshot of the terrain from an arbitrary viewpoint. The snapshot is saved in the project folder as an image file. The VR pointer command is a toggle button. The user can activate and deactivate a ray originating from the tip of the controller. The primary purpose of the ray is to provide cues for shared understanding and communication between the user in VR and the desktop user. The immersed user utilizes the ray to point at objects or features on the map. Additionally, the ray is used to query meta information of terrain positions or

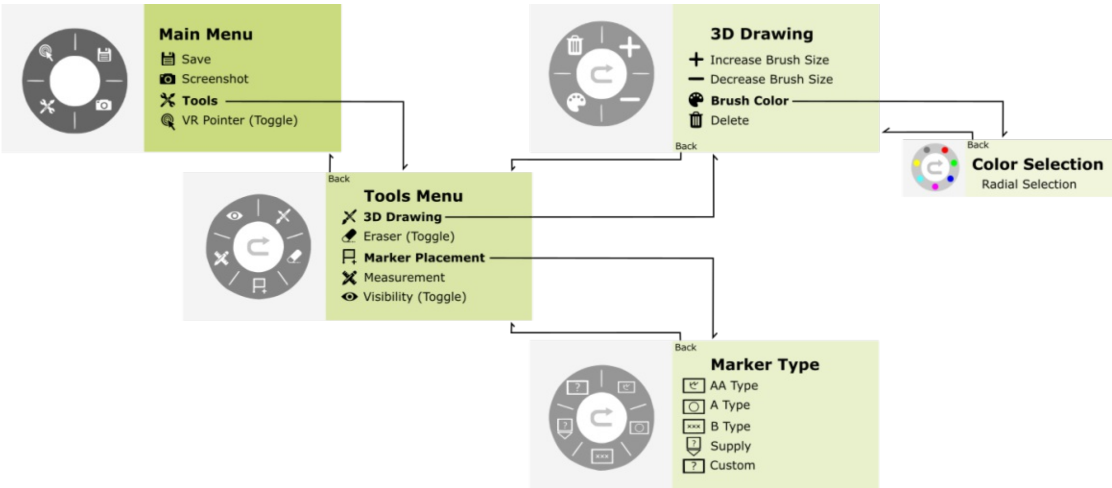


Figure 4.15: The hierarchy of the system radial menu in VR.

annotation. If the ray hits an object, the user in VR can read the current UTM position of the hit point and other quantities related to the hit object, such as circumference, area, and text labels. To avoid visual clutter, we decided to keep the design of the VR pointer simple. If the laser is active, the transparency of the ray gradually increases from the start point, and the ray is colored in a gray shade, as seen in Figure 4.16a. If the user hits the terrain with the ray, the ray is colored in a blue shade, and the hit point is highlighted. The label element on the controller shows the UTM coordinate of the hit point, seen in Figure 4.16b. Figure 4.16c shows the design of the ray if the ray hits a terrain object.

### 4.3.2 3D Tools

The tool command in the main level of the system controller radial menu invokes a submenu providing additional tools to manipulate and create objects in VR. The tools menu provides functions for 3D drawing, annotation removal, marker placement, measurements, and visibility analysis. Some spatial functions are not available in certain navigation modes. For instance, 3D drawing is not reasonable in the first-person navigation but


Figure 4.16: The design of the pointer in VR changes depending on the type of the target hit by the the ray emerging form the tip of the system controller.

useful in the table mode. The following paragraphs explain the implementation of the terrain tools available in VR in detail.

#### Eraser

To remove unwanted terrain annotations, the user in VR can activate the eraser command in the tools menu. As soon as the eraser is active, the interaction sphere is on top of the controller is enabled. If the sphere collides with an object in the scene, the color of the sphere changes, and the user can press the trigger button to erase the currently selected object. The eraser toggle command is an exclusive function. If the eraser is activated, all other commands are deactivated to avoid multiple assignments to buttons on the controller; for instance, the trigger button should be used for only one action at any given time. The user has to deactivate the eraser command to use others.

#### 3D Drawing

The primary purpose of the 3D drawing is to leverage the 6DoF interactions for terrain annotations. The officers can annotate the map directly using the controller. The tool for 3D drawing can be activated on the system controller in the tools menu. The interaction sphere at the top of the controller serves as a painting brush, and the size of the sphere is adjustable, as seen in the second level of the hierarchy in Figure 4.15. Pressing the plus symbol increases, and the minus symbol decreases the size of the sphere, which serves as the width for the 3D line. The user can then draw a polyline by pressing the trigger and moving the controller simultaneously. Figure 4.17a shows the output from the 2D and 3D drawing from the perspective of the desktop and immersed user. While annotations created by the desktop user are placed above the ground, the 3D lines are freely placed. Figure 4.17b, shows the same annotations in Figure 4.17a, but from the perspective of the desktop user. Every line, area, point, and marker is accompanied by an invisible 3D mesh used for collisions. One example is shown in Figure 4.18c.

We implemented a simple algorithm for 3D drawing. We sample points along the drawing motion as long as the immersed user presses the trigger during drawing mode. If the distance between the last recorded sample point and the current sample is larger than a fixed threshold, we add the current sample point to the current line. We use


Figure 4.17: Annotations viewed from different perspectives. **a)** 2D and 3D annotations from the perspective of the immersed user, **b)** from the perspective of the desktop user and **c)** shows the collider of a 3D line.

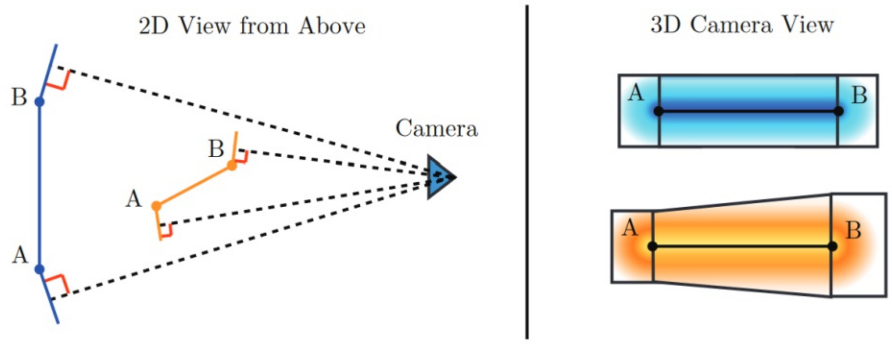


Figure 4.18: A 3D line consists of three quads. The rotation of the quads on the side depends on the camera's orientation, and the center quad rotates around the line direction [Hil12].



volumetric lines for the representation of the 3D lines based on a Unity3D plugin for line rendering [Unt21]. The plugin utilizes the GPU to extrude the geometry of a line in the vertex shader, based on an algorithm proposed by Sébastien Hillaire [Hil12]. A triangle strip presents a line, and one line in a polyline consists of three quads, shown in Figure 4.18. The rotation of the quads depends on the orientation of the VR camera. The quads at the end of the line always face the camera, and the quad in the middle only rotate around the direction of the line. The brush size, defined by the user in VR defines the width of the quads.

### Measurements

Like the desktop user, the immersed user can measure the linear distance between two given points using the measurement command in the tools menu. As soon as the measurement command is activated, the user has to choose two points on the terrain using a ray emerging from the controller with the trigger. The design of the ray is similar to the VR pointer. Figure 4.19 shows a measurement element from two perspectives,


Figure 4.19: A measurement object indicating the distance between two points in 3D. A measurement object is shown **Left:** from the perspective of the immersed user and **Right:** from the perspective of the desktop user.

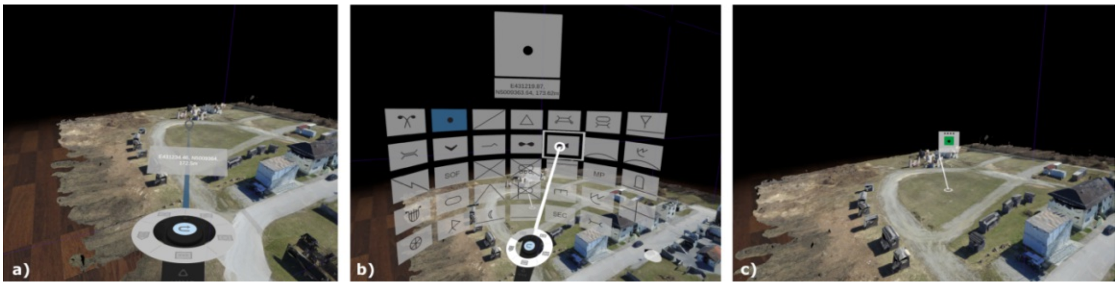

Figure 4.20: The marker placement process. **a)** The user can select a location on the terrain using a ray. **b)** All available marker symbols are arranged on a curved surface, and the user can select a symbol using a ray and the trigger button. **c)** A placed marker from the perspective of the immersed user.

one from the viewpoint of the immersed user and one from the desktop user. The text label in VR is a billboard facing the VR camera. The measurement element placement is exclusively available to the immersed user. The desktop user can query the distance between two points using the drawing tool. Placed annotation elements can be selected inside the map view on the desktop, and quantitative properties, such as length and circumference, are shown in the context view at the bottom of the desktop screen.

### Marker Placement

The immersed user can place military markers using the command in the tools pie menu. As soon as the command is triggered, the user has to choose a location on the map using a ray, seen in Figure 4.20a. While hovering the ray over the terrain, the user sees a preview marker at the current ray hit point and the respective UTM coordinate. If the user utilizes the trigger, the current hit point is chosen, and the user can edit the elements of the marker symbols. The radial sub-menu shows options for all marker symbol types described in the previous section: unit, base, size, supply, or custom image elements. We automatically arrange the image selection of all available symbols on a curved surface in front of the immersed user grouped by image type, seen in 4.20b. The user can then utilize a ray emerging from the controller tip to select a symbol and confirm that selection using the trigger button. Above the curved surface, the user can see a preview of the current marker to be placed. The back button in the center of the touchpad confirms the marker placement. The placed marker is visible from the desktop as well as in VR, seen in 4.20c. A line from the bottom left corner of the marker to the terrain indicates the exact location on the ground. The image rotates towards the gaze of the immersed user, so the marker symbol is visible from any position and orientation.

### Visibility Analysis

One of the main advantages of the immersive geographical system is easy accessibility to quantitative and qualitative properties of the terrain structure. In the context of

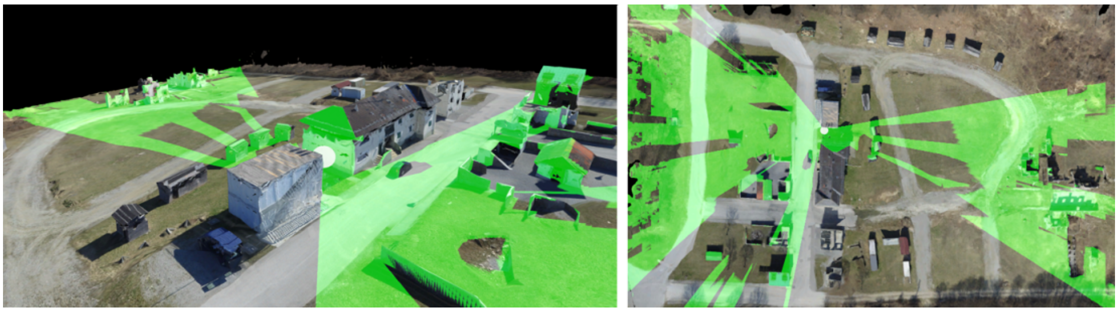

Figure 4.21: The result of the visibility feature is a color-coded terrain texture. Areas highlighted in a light green shade are visible from the position of the white sphere. **Left:** The line-of-sight analysis perceived from the user in VR and **Right:** from the desktop user.

military staff training, we aim to provide decision-makers a visibility analysis tool that highlights visible and non-visible areas from a given location. This prototype implements a visibility tool available in the VR table mode. As soon as the user in VR activates the visibility toggle button, a white sphere appears on the terrain. The user can adjust the sphere's position by dragging the object using the controller and the trigger button. The surrounding terrain visible from the position of the sphere is highlighted using a light green shade, representing the positive line of sight from this position, as shown in Figure 4.21. Areas without a green shade are not visible from the current position of the sphere. For the visibility feature, we use a custom terrain shader and a light source. The white sphere represents the origin of a point light, and in the fragment shader, we access the shadow map cast by the point light. If the current fragment is visible from the position of the light source, a bright green color is added to the color of the fragment.

CHAPTER 5

# Results and Evaluation

In this chapter, we present the visual and functional results of the use cases discussed in the design chapter and final remarks from our project partners at IMG. We further report the technical performance of our prototype. Lastly, we elaborate on the limitations of our framework and possible improvements.

## 5.1 Qualitative Results

This section represents the main outcome of this work, namely the implementation of our VR prototype aiming to provide support for mission planning tasks and decision-making at a military staff training.

### 5.1.1 Implemented Use Cases

In chapter 3, we derived two use cases from our observations on a military staff exercise: I) terrain annotations and II) visibility and quantitative analysis. The following paragraphs compare the concept design of the use cases discussed in the design chapter of this thesis, with the resulting prototype and its features.

#### Terrain Annotations

In our prototype, the virtual environment serves as a task space for tactical planning. Users add information related to their strategy, reason from others' visual instructions, and decide about future tactics. More than one user can annotate the terrain and work on a given task simultaneously. The objective of this use case is to support officers during the planning phase by providing tools to annotate the terrain. We implemented commands for 2D and 3D annotations to meet this goal. Figure 5.1a and Figure 5.1c show example annotations on a paper map of a tactical strategy during military staff training. Military experts drew areas, lines, NATO markers and added text-based information on a

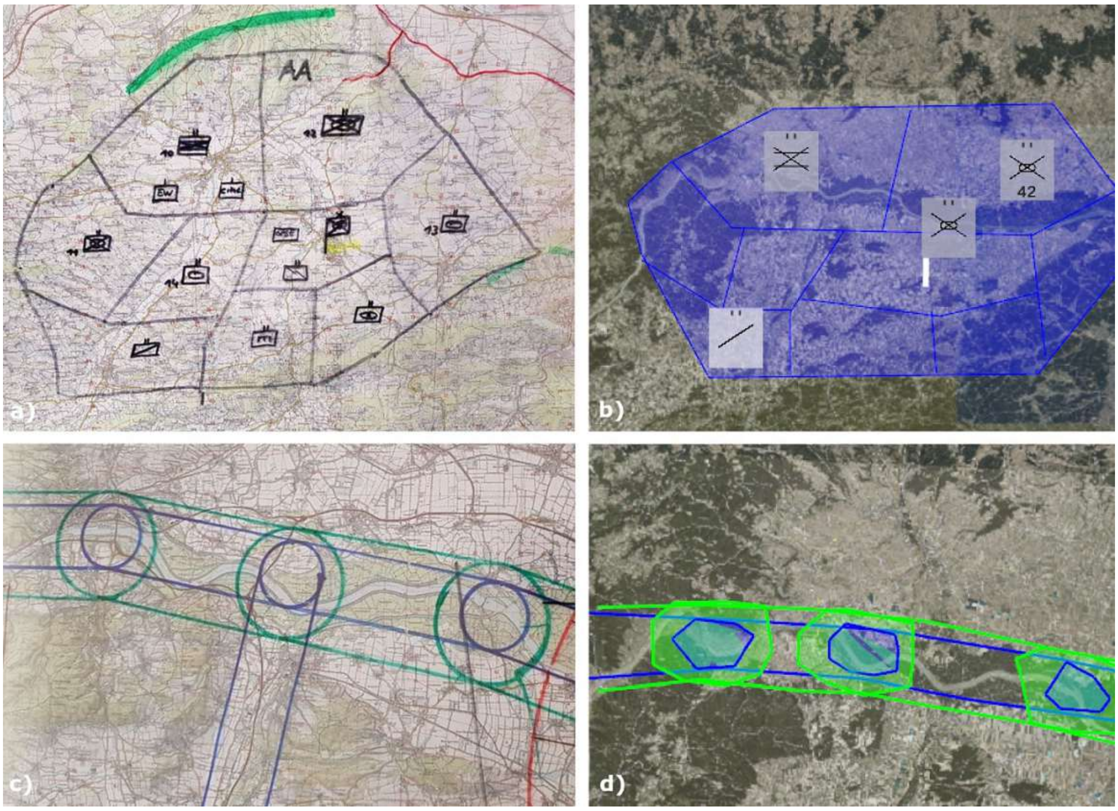


Figure 5.1: **a) and c):** During the planning phase of a military staff training, tactical strategies are drawn on a transparent sheet and overlaid on a paper map. **b) and d):** The essence of the tactical strategies realized using our VR prototype.

transparent sheet overlaid on a paper map. Participants of such a military staff training can render the same output using our prototype, seen in Figure 5.1b and Figure 5.1d. Both users, the desktop user and the user in VR can use the 2D and 3D drawing tool to annotate 3D digital terrain data.

We implemented several features providing additional user guidance for the mission planning process and supporting terrain annotations placement: I) communication cues for user collaboration and II) dynamic viewpoint selection.

**Communication Cues for User Collaboration:** a key attribute of an effective collaborative work is efficient communication. We implemented several awareness cues as a means of non-verbal communication. The following features aim to support collaborative work:

- **VR-Pointer:** The user in VR can use a virtual ray to select and point at specific locations or objects, seen in Figure 5.2b. Our observations of the field study inspired this feature. During presentation mode, the commander pinpoints positions on the paper map and orders the officers to alter their strategy, investigate certain

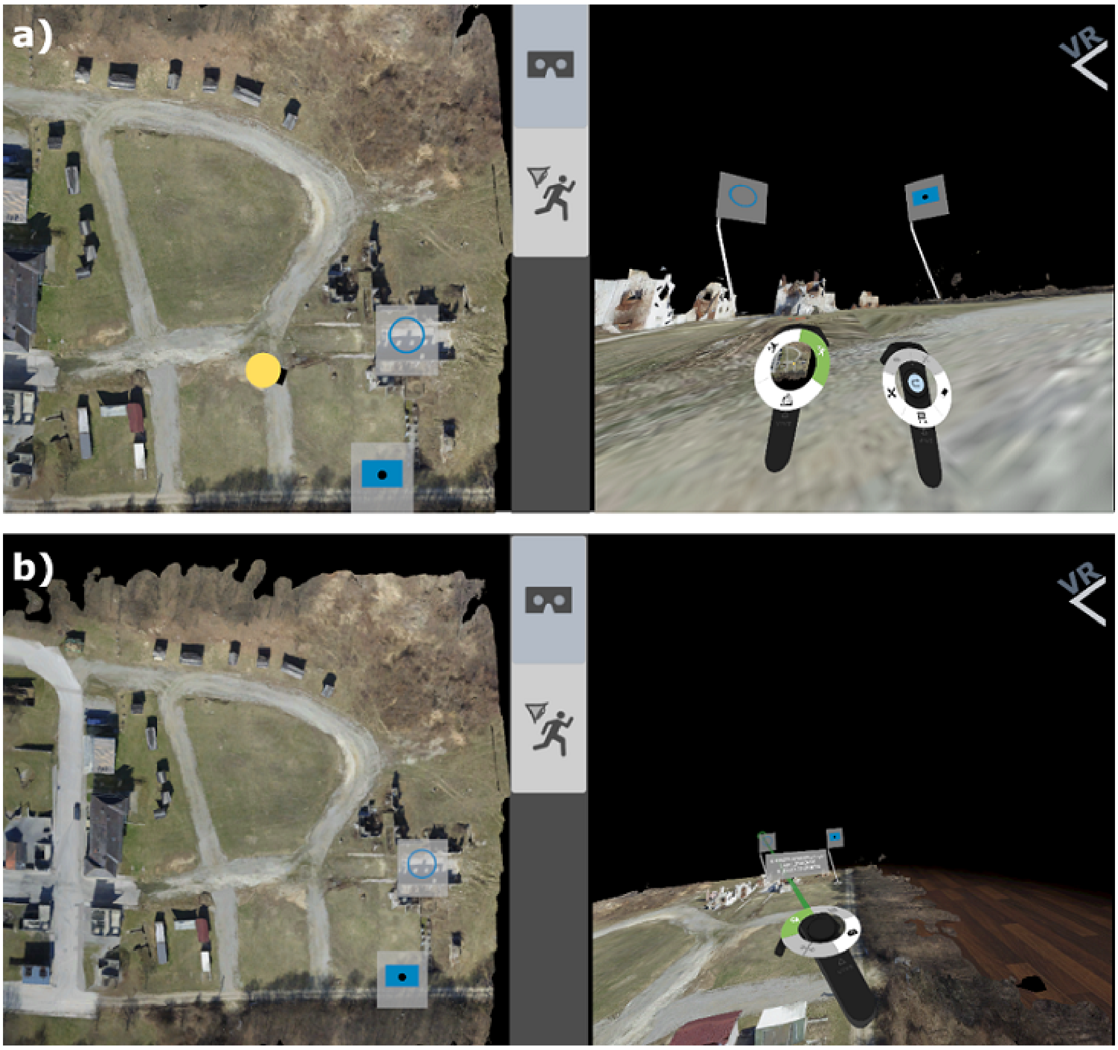


Figure 5.2: We implemented several features to facilitate communication and awareness during collaborative work. In image **a)**, the desktop user can observe the position and gaze of the VR user in first-person navigation mode. Image **b)** shows the VR pointer, a communication tool for the user in VR.

circumstances, or inquire more information on data related to the mission. We argue that the pointer in VR is helpful for collaboration by providing non-verbal communication from the immersed user to the others watching over the actions of the immersed user on the desktop.

- **Gaze-View:** The user on the desktop can observe the actions of the immersed user through an adjustable window on the desktop, shown in Figure 5.2a and Figure 5.2b on the right. Our software offers two perspectives; a third-person point-of-view, and the first-person point-of-view, showing the gaze direction and the field-of-view of the immersed user to other users outside of the virtual environment.

- **VR Avatar:** We placed an avatar onto the terrain in the map view on the desktop, seen in Figure 5.2a. If the immersed user navigates the scene using free-steering or first-person mode, the avatar's position is shown on the desktop map view for the desktop user. This way, the user on the desktop can see and analyze the current position and orientation of the immersed user on the map.
- **Selection:** We highlight selected annotation elements in the scene to indicate objects which are currently used by the user.
- **Real-time Feedback:** The changes made in the scene, such as annotations or disabling/enabling annotation layers, are visible to all users, concurrently.
- **Post-editing:** We allow the desktop user to rework placed markers. This allows for coarse placement of markers in VR and refinement on the desktop. This way, two users can split a task when working collaboratively. Other types of annotations, such as points, can not be adjusted after placement, but can be deleted.

**Dynamic View-point Selection:** For terrain exploration and annotation, an essential feature of the prototype is navigating the environment in VR in a meaningful way. We decided to solely dedicate a VR controller to commands related to locomotion to provide the immersed user a straightforward interface for navigation. Furthermore, the user can explore terrain data from different viewpoints and adjudicate spatial structures and terrain properties. With different navigation modes, the prototype provides unique insights into terrain conditions. Figure 5.3 presents the navigation modes on different terrain types. These modes are designed to deal with different types of terrain properties. The first snapshot in Figure 5.3a shows a coarse reconstructed area with an extent of 76600$m$ x 69102$m$ x 578$m$. The first-person mode is deactivated due to the extent of the terrain. A first-person view is not an advantage on rough reconstructed terrain and might lead to disorientation. The second image in Figure 5.3b, shows a fine reconstructed

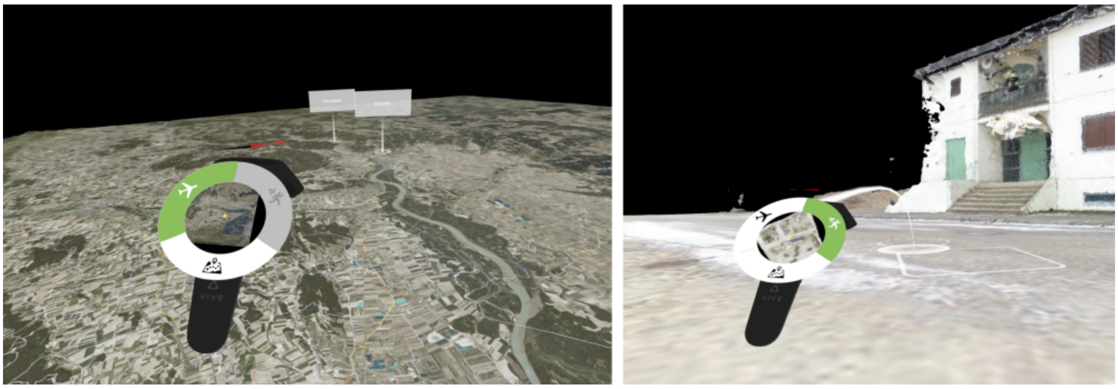

Figure 5.3: The user in VR can navigate the terrain using different means of locomotion. **Left:** The table mode, including landmarks for orientation, and **Right:** the first-person mode.


Figure 5.4: The visibility tool helps to analyze line-of-sight from a chosen position. **a)** The sphere is placed in the corner of a building. Visible areas from the position of the sphere are colored in a green shade. **b)** If the sphere is placed inside a building, nothing outside the building is visible. Image **c)** shows that from the current position of the sphere, people would not see the bottom of the house on the left side.

terrain in a first-person view. The miniature map of the terrain in the center of the navigation controller shows the current location of the immersed user. Teleportation is advantageous for fine reconstructed terrain, like urban areas. The user in VR can inspect reconstructed buildings and structures closely. This way, the officers and commanders can include information about the local infrastructure of mission sites in their decision-making process.

### Visibility and Quantitative Analysis

Common tasks during the military staff training on paper maps are distance estimation and line-of-sight analysis. The developed prototype provides an intuitive way to analyze visibility from a given viewpoint and extract quantitative measurements, such as distances or areas, in an immersive virtual environment. Figure 5.4 depicts the implemented visibility feature on different positions on the terrain. The user in VR can choose an arbitrary 3D position as the origin for visibility analysis by adjusting the position of a white sphere. In Figure 5.4a, the sphere is placed behind an l-shaped building, 1.90$m$ from the ground. The desktop user and the immersed user see the line-of-sight from that position and can adjust their tactical strategy for the given exercise task accordingly. If the sphere is placed inside a building, as shown in Figure 5.4b, nothing is shaded in a green color except the building. Figure 5.4c depicts one advantage of the visibility feature. The sphere is placed next to a building, 1.90$m$ from the ground. A person with an eye-level of 1.90$m$ would not see the bottom of the tiny house on the left from that position. Figure 5.4c is taken from the perspective of the immersed user and Figure 5.4c and Figure 5.4c from the desktop user.

In addition to visibility analysis, the prototype provides features for quantitative measurements. Both users can query properties from terrain objects, such as annotations or the terrain itself. The immersed user can utilize a virtual ray, provided in the system menu controller's radial menu, described in section 4.3.1 and shown in Figure 5.2b, to retrieve the geolocation of terrain points or object properties, such as circumference and area of a region-of-interest or length of an annotation line. Another tool to measure distances in VR is the measurement tool shown in Figure 5.2c and Figure 5.2d, from the

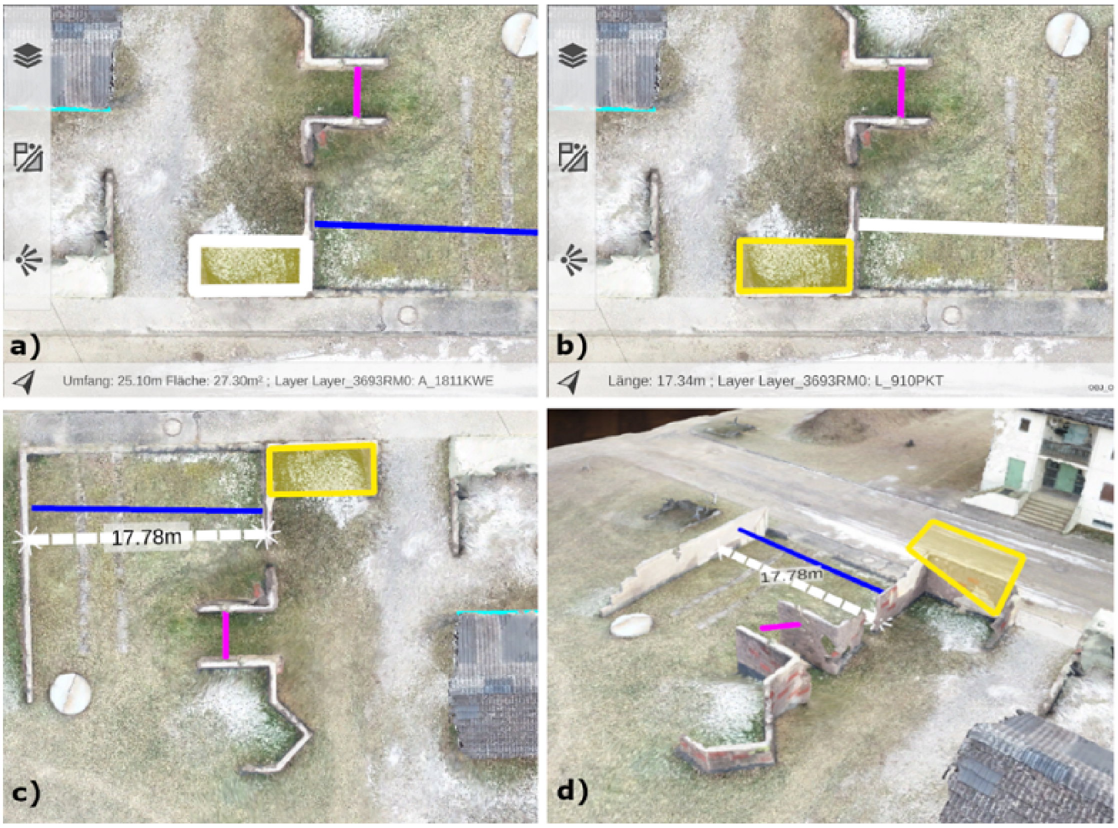


Figure 5.5: The desktop user and the immersed user can use several tools to retrieve the quantitative properties of scene objects. Image **a)-b)** show the retrieval of quantitative information from the desktop and image **c)-d)** depict a white ruler, the measurement tool provided for the immersed user.

perspective of the desktop user and the immersed user, respectively. Figure 5.2a and Figure 5.2b show how the desktop user can retrieve quantitative properties of objects. If the user selects an object, for instance the yellow rectangle in Figure 5.2a, the properties of that element are shown in the context view on the bottom of the screen in Figure 5.2a or Figure 5.2b. Selected annotation objects are colored in a white shade.

### 5.1.2 Expert Feedback

A domain expert in military geography, who guided us through the design and development process, tested our prototype with other experts from the geography department, and showcased the implemented features to a brigade group, consisting of officers in training and commanders at a military staff training. Subsequently, we conducted an unstructured interview with the expert. Our objective was to investigate the potential of VR-based tools to support mission planning and decision-making using our prototype. We asked about their expertise, common tasks, observations, conclusions, and observed advantages and limitations of the prototype compared to alternative tools. In the following, we present the most paramount insights gained form the interview.

Each officer participating in a staff training is specialized in a certain domain. According to the expert, the responsibility of an officer trained in geography is to consult the leading commander in regard to the mission ground. Their tasks include: identification of potential problems about the mission area; for instance, weather condition, political climate, and foresightful supervision of the current mission planning events to provide planning material for the commander and officers promptly. The geographer has to precisely understand the staff training procedure to support other participating officers. Their goal is to think about information the commander might need for decision-making and to prepare that information rapidly with the help of technical analysts. The expert describes the motivation to use VR as a support tool for the commander as follows: *"With the help of the prototype, we are able to use the third dimension. The problem with a paper map is, that you have to read it correctly and imagine what the terrain will look like. Unfortunately, fewer and fewer people can read and understand a terrain using contour lines or hill shades."*

The expert assessed several features to be in particular helpful: The annotation layers, texture layers, visibility analysis tool, and terrain measurement tool. The subject of one staff training, where the prototype was tested, was the massive explosion in Beirut's port [BQW20] in 2020. During the staff training, the officers had to evaluate the implications of the explosion on the surrounding area. The officers trained in geography prepared the 3D data for the affected port area, with two textures. One texture showed the port before the explosion and the other one included notes and marks about the damages around the explosion epicenter. The objective was to rate the damages and prioritize areas for cleaning. Our interview partner mentioned that the prototype allows showing the commander the essence of the analysis in an intuitive way. Another example of an essential task, according to the expert, is to measure the dimension of streets. For path-finding and planning, officers and commanders in-charge have to know if certain support vehicles fit on a bridge, in a tunnel, or on rural roads along a planned path. As reported by the expert, the annotation layers help group a situation analysis into information layers. While one annotation layer includes terrain measurement elements, another show the route map for the vehicles. The layers can be disabled and enabled by the desktop operator, depending on the current analysis of the commander in VR.

An early version of the prototype was also tested at a military staff training focused on defense. The objective was to seal the perimeter of a building in an urban area from a hostile group and protect their troops. Two groups of officers were in-charge of planning the positions of the road barricades and railroad embankments. Both groups had the option to use our prototype for decision-making during the planning phase. One group utilized the prototype to evaluate their plan and investigated the positions of the embankments using our visibility analysis tool. With the help of the tool, they recognized that they have to change the placement of some railroad embankments, otherwise a hostile agent has line-of-sight access to their troops. Another advantage of the visibility tool, explicitly mentioned by the expert, is the retrieval of covered areas to approach a hostile group.

The interviewed domain expert mentioned not only success stories, but also behavioral observations. They talked about the attitude of some users towards the prototype according to age. Older officers tend to skip the prototype for decision support but order their younger colleagues to investigate certain circumstances regarding the mission objective. According to the observations of our interview partner, the younger generations tend to be more experienced when using VR applications and are more eager to use the technology. Regarding the potential of VR, the expert said: *"The main tool used to be the paper map and now the trend moves towards digital tools, for instance, 2.5D HTML models or, as in the prototype, 3D models"*. Even though most of the officers were interested in trying our prototype, there are certain drawbacks, in comparison to paper maps.

VR-based tools need energy, physical space, and the right room conditions, like technology-friendly room temperature and low humidity. A paper map works anywhere, for instance in a tent, is easy to transport and replication is cheap. However, only experienced and skilled officers can read a paper map and derive information reliably. According to the expert, a user in VR can understand complex circumstances and terrain faster, and derive positions more precise. For instance: *"In the virtual environment you have a one to one terrain representation. If there is a command post marker, we can see the position in centimeter accuracy. On a paper map, depending on the benchmark and extents, a marker can be 10m or 100m far away from its exact location due to the limitation of the paper map resolution. To read the exact location of a marker on a paper map, you have to use your imagination, but VR can show it well."*

At the end of our interview, the domain expert suggested some improvements for future work. One idea regards the implementation of the 3D drawing. The expert argued that direct drawing on the terrain or on certain height-level above the terrain would be more helpful for planning tasks. Furthermore, placed annotation elements, such as markers, should scale dynamically based on the distance to the user in VR to avoid occlusion. The most important feature request is an automated pipeline to import ground and terrain data to the prototype directly from GIS software used by the military geography department. Further, the interviewee suggests that annotation layers of textures layers should be imported at run-time to allow prompt changes and adaptions. For instance, a commander can stay in the immersive environment, while an officer can prepare new terrain data, textures, and annotations. The newly prepared data is then imported into the prototype at run-time without the need to restart it.

## 5.2 Technical Evaluation

In this section, we outline the technical performance of our VR prototype. We evaluated the processing time of the geographical data import and the system's overall performance while processing resource-intense computations. For our tests we used the following

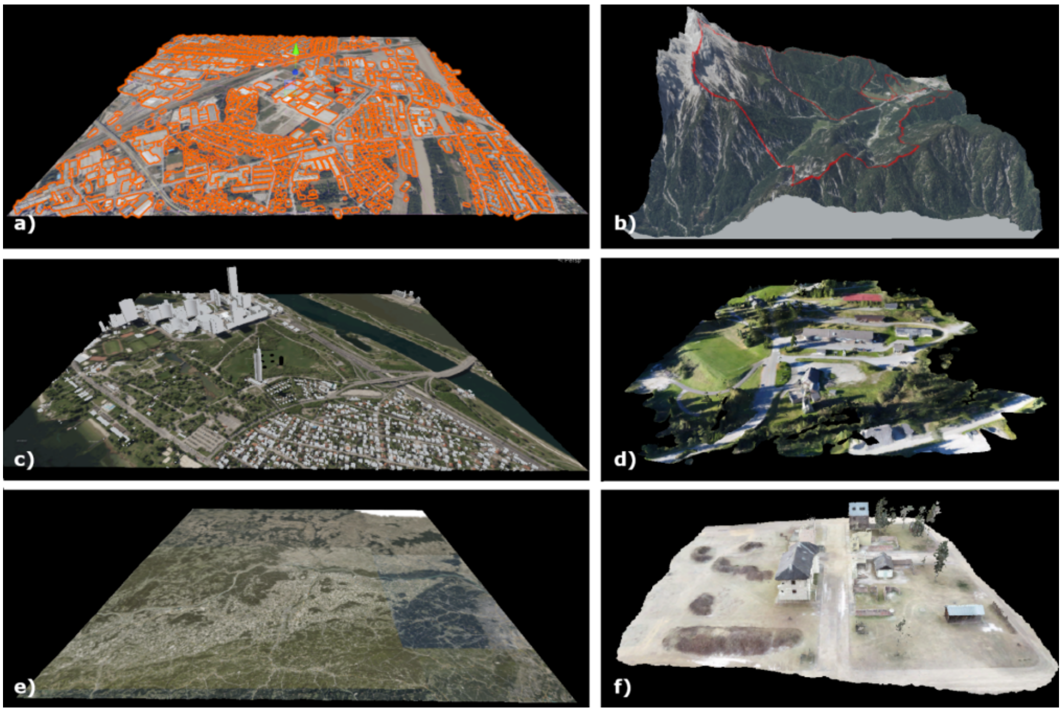


Figure 5.6: Terrain test data obtained from the IMG. **a)-c) and e)** are based on DEM models and **d) and f)** are reconstructed using aerial imaging.

hardware: a desktop PC running Windows 10 Pro, Nvidia RTX2080 GPU, an Intel(R) Xeon(R) CPU E3-1245 V2 @ 3.40GHz CPU and 32.0 GB Random-Access Memory (RAM). VR applications demand specific requirements on the computer hardware for preventing cybersickness and providing a comfortable user experience. Accordingly, we aim to keep the frame-rate above 90 Frames Per Second (FPS) at all times. The following sections present the terrain test dataset used for testing the performance of the prototype.

| | **# of Triangles** | **File Size (MB)** | **Texture Size (MB)** | **Import Time Total (s)** |
|---|---|---|---|---|
| File a) | 647.880 | 71.46 | 11.86 + 47.58 | 9.85 |
| File b) | 727,297 | 74.01 | 2.83 | 7.40 |
| File c) | 302,029 | 28.77 | 2.09 | 3.77 |
| File d) | **1,143,750** | **129.37** | 30.00 | **26.5** |
| File e) | 793,272 | 83.15 | **95.19** | 16.30 |
| File f) | 752,708 | 87.85 | 5.49 | 16.20 |

Table 5.1: List of selected terrain test data files, their properties and import times.

### 5.2.1 Terrain Test Data

The IMG provided us with several terrain data files to test the overall implementation of our prototype. The data varies in shape, extents, triangle count, texture resolution, triangle mesh resolution, topology, and mesh composition. We used the Unity3D library TriLib to import the geographical dataset, including textures. Table 5.1 shows different test files, their properties, and the import time in seconds. The import time is dependent on the number of mesh triangles, the image resolution of all imported textures, and the number of terrain textures.

### 5.2.2 Performance Analysis

One essential technical requirement was to achieve a minimum of 90 FPS for the user in VR at all times. During the development, we regularly tested the performance of our prototype to investigate potential bottlenecks. Using our technical setup and the given terrain test data, we did not encounter significant performance interference, except while using the tool for line-of-sight analysis. The visibility tool was the most consuming calculation executed by the prototype, and the performance heavily depended on the triangle count of the terrain data. For the line-of-sight calculations within the visibility analysis, we use Unity3D's internal shadow rendering. Based on our observation, a terrain triangle count of 500k could already reduce the frame rate to 45-50 FPS. This might lead to an increase in discomfort for the user in VR. Therefore, we recommend a more elaborated data import pipeline for future prototype iterations, such as tiled terrain data.

## 5.3 Limitations and Improvements

This section critically reflects on the results of our work, discusses limitations surfaced during the implementation, and briefly suggests possible technical and design improvements.

### 5.3.1 Occlusion and Visual Clutter

A common effect observed when working with visual information is *visual clutter*. Visual clutter refers to a high density of visual elements in one place [RLN07]. Visual clutter usually leads to a high cognitive load, occlusion of other visual elements, and less effective interfaces due to decreased readability and interpretability of visual information. We implemented several methods to avoid occlusion of information:

- **Layer Transparency:** The transparency of annotation layers is adjustable. Users can disable annotation layers or influence the degree of the overall visibility in the case of occlusion.
- **Dynamic Placement:** One method to avoid visual clutter is to allow dynamic scaling, rotation, and placement of annotation elements depending on the user's viewing direction. In our prototype, the desktop user can adjust the position of a

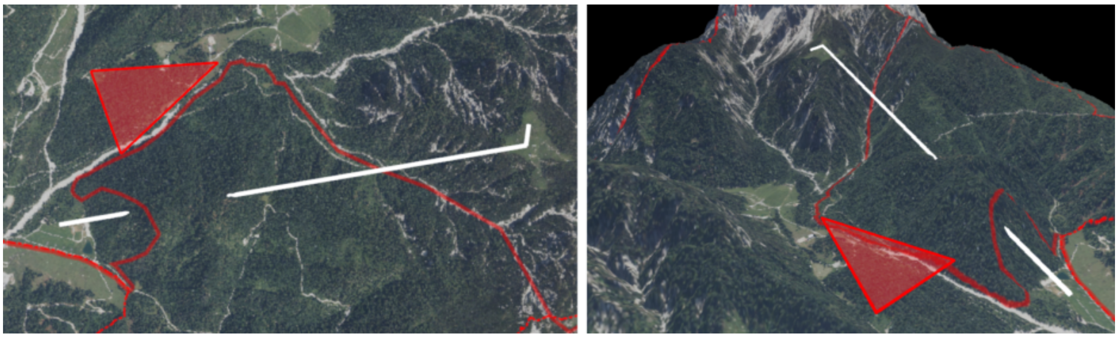

Figure 5.7: Occlusion is a common problem when visualizing different layers of information at once. Due to the topology of the terrain data, part of a line, shown in white, is occluded. **Left:** From the perspective of the desktop user and. **Right:** From the perspective of the immersed user.

desktop marker to avoid occluding other elements or terrain information (seen in Figure 5.8). Nonetheless, the current implementation of the design of the annotation elements could be further improved.

- **Disc Pointer:** We designed the tip of the ray in VR, so it does not occlude information on the terrain. We used a flat disc in transparent white gradient color.

Some occlusion persists because avoiding it is particularly challenging in a 3D environment. Depending on the viewpoint of the immersed user or the terrain curvature, terrain data can occlude annotation elements. One example is shown in Figure 5.7. The desktop user placed a line on the terrain, and part of the line is not visible due to the topology of the terrain data. Furthermore, if the immersed user navigates a fine reconstructed environment from a first-person perspective, such as an urban area, buildings, or other ground elements can occlude annotation elements. A possible solution would be a see-through material for the terrain mesh or dynamic placement of marker billboards and other annotation elements, depending on the user's viewing direction and line of sight.

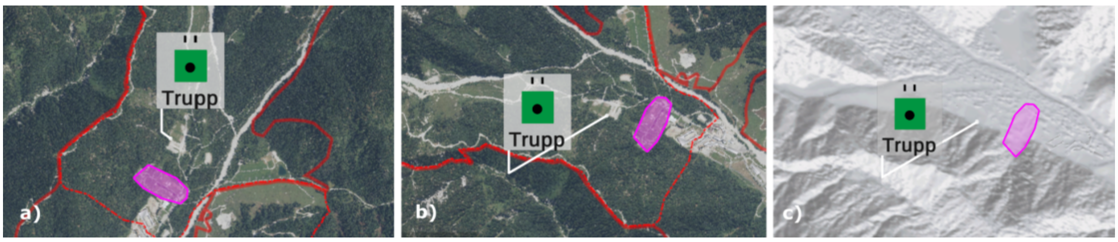


Figure 5.8: The desktop user can adjust the position of a desktop marker to avoid occlusion. **a)-b)** Marker elements do not rotate with the map view. A white line indicates the exact location of a marker. **c)** The terrain texture can be disabled, to reduce visual clutter.

### 5.3.2 Data Management

Our prototype provides unique viewpoints on terrain data and tools to measure quantitative properties in a multi-user setting. It is important to highlight that the overall accuracy of measurements and visibility calculations strongly depends on the level-of-detail of the terrain, correct alignment and preparation in Blender, and correct definition of geographical properties, such as scale benchmark. For instance, coarse reconstructed data is not ideal for visibility analysis due to missing terrain structures, such as buildings, streets, woods, or hills. We implemented means to disable user commands not suitable for the type of imported terrain. Our prototype deactivates certain functions, such as visibility analysis or first-person navigation, if the extent of the terrain is greater than a certain threshold. The wider the extent of the terrain, the less accurate are our calculations. For instance, terrain data with the largest extent being $50km$, and the tracking volume of the VR space is $2m \times 2m$, then $1m$ in table-top navigation mode is $25km$ on the terrain. A slight hand movement, such as $10cm$ distance, is then $2.5km$. If the user navigates a broad terrain in the first-person mode, we have to scale the immersed user to achieve a terrain-user ratio of 1:1. Broad terrain can lead to a very small scaling factor and, therefore, numerical issues. Another limitation related to the terrain data is triangle count. Some calculations executed in Unity3D linearly increase with the number of mesh triangles, such as the visibility analysis tool.

Overall, our prototype's usability depends on the correctness of the data preparation, accuracy of the terrain reconstruction, and the extent of the terrain data.

CHAPTER 6

# Conclusion and Future Work

In this thesis, we presented a VR prototype as a digital support tool for collaborative planning tasks and decision-making. The design and functionalities of the prototype are based on state-of-the-art research and tailored to an Austrian military staff exercise. We elaborated on the conducted field study, our observations, and derived requirements on function and design. Further, we explained the implementation process and the motivation behind our design and development. We followed a participatory design approach, by working closely together with domain experts to further understand the use case and continuously improve the interaction techniques and functionalities.

Initial field tests demonstrated a great potential for VR-based decision tools in military staff exercises. According to our interview with a domain expert at the IMG, the functionalities of the prototype, such as the collaborative aspect, immersive viewpoint, 3D interactions with the terrain, as well as the line-of-sight analysis, offer qualitative advantages compared to analog alternatives. Initial deployment showed that commanders and officers are interested in the prototype due to its novelty for the field and potential to reduce the duration of the planning process. However, digital tools can never entirely replace paper maps due to their low adaptability and reliability. Digital tools depend on the infrastructure of a mission site, such as energy, which is not always accessible. Our results showed that the prototype can be used as an additional tool to analyze complex terrain structures during the planning process and decision-making of a training session. The application is currently showcased at different departments of the Austrian army and might be integrated in the training curriculum for officer candidates.

Unfortunately, due to the COVID-19 pandemic, we were not able to conduct a formal user study to evaluate the effectiveness of our user interactions and overall design. Instead, we conducted an unstructured interview with a domain expert to investigate the potential of our VR-based prototype in military staff training sessions. Our interviewee tried the prototype in the on several occasions and discussed their observations with us. The domain expert concluded that our prototype is especially useful for officers to understand the terrain and line-of-sight. Commanders mainly use the prototype to assess tactical strategies in VR prepared by the officers.

Along with great potential, the interview showed room for improvement. One limitation of our prototype is that the quality and accuracy of quantitative measurements and visibility analysis on the terrain heavily depends on the accuracy of the imported data and mesh resolution. In future work, the investigation in automatic mesh tiling, floating terrain origin and support for terrain level-of-detail might prove important to overcome issues with large terrain data. The next iteration of the prototype should focus on reduction of visual clutter and the implementation of more advanced methods to avoid occlusion in VR. Such approaches could include dynamic placement and scaling of scene elements, based on the immersed user's position and gaze, or dynamic terrain shaders revealing occluded information.

# List of Figures


1.1 **Right:** Example scene of a military mission training process. Source: Österreichisches Bundesheer/Rene Auer **Left:** Representation of a standard 2D paper map used during mission planning and presentation. . . . . . . . . 2

2.1 An Example of proprietary GIS software, **Left:** Mapbox Studio, a framework for creating custom maps [Map21], **Right:** ArcGIS - the 3D map indicates the walking distance to the nearest public transport in San Francisco using color code [Esr21]. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
2.2 Two examples of immersive virtual geographical environments. **Left:** A CAVE with three walls [SB07], **Right:** A person using a state-of-the-art VR HMD [CMH16] . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
2.3 Different navigation modes in Google Earth VR. Images courtesy of [iGEV21]. **Left:** Teleportation using ray-casting. **Center:** Users utilize the world map in the left hand to choose a location. **Right:** Text-based search and travel. 12
2.4 Space and time taxonomy of collaborative systems [NLSB19] [TBG05] . . 15
2.5 An abstract setup of a remote and on-site immersive collaborative virtual environment. The difference is the physical location of the users utilizing the virtual environment. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
2.6 Proposal of a high-level conceptual staff exercise structure. The figure displays the relationship between the crisis scenario and crisis prevention staff. [HRL14] 19
2.7 **Left:** An example of a tactical situation report in 2D space [MK16], **Right:** A digital sand table for tactical situation analysis in 3D space [AG00] . . 20

3.1 Example scene of a military staff training, Source: Österreichisches Bundesheer/Rene Auer . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
3.2 The timeline of the observed staff exercise . . . . . . . . . . . . . . . . . . . . 24
3.3 Data and information flow of the proposed application. . . . . . . . . . . 28
3.4 The HTC Vive Pro components. From left to right: the head-mounted display itself, the input devices, also called controllers, the lighthouse sensors used to track the pose of the headset and the controllers, and a wireless adapter. Images taken from the official HTC website [Cor21a] . . . . . . . . . . . . 30
3.5 UI concept of the desktop interface based on the requirements. . . . . . . 32

3.6 **Left:** The two input controls used for spatial interactions, the trigger an the touchpad. **Right:** The touchpad serves as an input for the radial menu in VR. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33

4.1 The hierarchy and components of the system of the proposed prototype. The architecture components can be grouped into three tiers: presentation, logic, and data tier. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38
4.2 A high-level summary of the object generation process. One script generates all other scripts at run-time to reduce manual adjustment in the Unity3D editor. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
4.3 The properties of the used terrain data vary strongly. **a)** A fine reconstruction of a mountain range, **b)** a wide map and coarse definitions of riverbeds and hills, **c)** terrain derived from DEM data, consisting of a base map and building data and **d)** a reconstruction of military exercise grounds, based on drone images. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
4.4 Flow chart of the processes at start-up. . . . . . . . . . . . . . . . . . . . 47
4.5 The arrangement and implementation of the UI elements. **Left Column:** An abstract overview of the main UI elements. **Right Column:** The implementation of the UI design. The top row presents the UI if VR is not enabled, and the bottom row depicts the arrangement of the elements after activation of the VR mode . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
4.6 The desktop user can choose between two different perspectives to observe the actions of the user in VR. **Left:** A first-person point of view and **Right:** a third-person point of view . . . . . . . . . . . . . . . . . . . . . . . . . 50
4.7 The desktop user can adjust the appearance of the annotation layers via the layer management window in the tools menu. . . . . . . . . . . . . . . . . . 51
4.8 The desktop user can adjust the appearance of the imported terrain texture layers via the texture management window in the tools menu. . . . . . . . . 51
4.9 The Material Capture (MatCap) shader is used to enhance terrain structure. **a)** The textured terrain, **b)** the reference texture encoding the lighting setup and **c)** the same terrain data rendered with the MatCap shader. . . . . . 52
4.10 The annotation window provides tools for drawing on the terrain and placing military markers. **Left:** The annotation window and **Right:** annotation elements on the terrain, such as points, lines, areas and markers, from the perspective of the desktop user. . . . . . . . . . . . . . . . . . . . . . . . . 53
4.11 The left three images present the structure of the three marker types used in this prototype: custom, composited, and supply marker. The two images on the right show an example of a composited marker. The desktop user can move the marker to avoid occlusions. The exact position of the marker is then indicated by a line from the marker position to the edge of the marker. . 54
4.12 The color design and spatial arrangement of the user interface elements in VR. **Left:** The color design of the radial menu. **Right:** The arrangement of the user interface elements. . . . . . . . . . . . . . . . . . . . . . . . . 56

4.13 The structure of the radial menu designed to help the user in VR to get around the virtual terrain. . . . . . . . . . . . . . . . . . . . . . . . . . . 56

4.14 The prototype provides three different navigation methods for the immersed user. **a)** The table mode. Image **a.1)** shows the avatar of the user on the terrain, and image **a.2)** shows the magnified view and the metadata, such as terrain coordinates, while the user holds the avatar. Image **b)** presents the first-person navigation using teleportation and image **c)** free steering navigation using the trigger. . . . . . . . . . . . . . . . . . . . . . . . . . . 57

4.15 The hierarchy of the system radial menu in VR. . . . . . . . . . . . . . . 59

4.16 The design of the pointer in VR changes depending on the type of the target hit by the the ray emerging form the tip of the system controller. . . . . . 59

4.17 Annotations viewed from different perspectives. **a)** 2D and 3D annotations from the perspective of the immersed user, **b)** from the perspective of the desktop user and **c)** shows the collider of a 3D line. . . . . . . . . . . . . . 60

4.18 A 3D line consists of three quads. The rotation of the quads on the side depends on the camera's orientation, and the center quad rotates around the line direction [Hil12]. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 61

4.19 A measurement object indicating the distance between two points in 3D. A measurement object is shown **Left:** from the perspective of the immersed user and **Right:** from the perspective of the desktop user. . . . . . . . . . . 61

4.20 The marker placement process. **a)** The user can select a location on the terrain using a ray. **b)** All available marker symbols are arranged on a curved surface, and the user can select a symbol using a ray and the trigger button. **c)** A placed marker from the perspective of the immersed user. . . . . . . 62

4.21 The result of the visibility feature is a color-coded terrain texture. Areas highlighted in a light green shade are visible from the position of the white sphere. **Left:** The line-of-sight analysis perceived from the user in VR and **Right:** from the desktop user. . . . . . . . . . . . . . . . . . . . . . . . . 63

5.1 **a) and c):** During the planning phase of a military staff training, tactical strategies are drawn on a transparent sheet and overlaid on a paper map. **b) and d):** The essence of the tactical strategies realized using our VR prototype. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66

5.2 We implemented several features to facilitate communication and awareness during collaborative work. In image **a)**, the desktop user can observe the position and gaze of the VR user in first-person navigation mode. Image **b)** shows the VR pointer, a communication tool for the user in VR. . . . . . 67

5.3 The user in VR can navigate the terrain using different means of locomotion. **Left:** The table mode, including landmarks for orientation, and **Right:** the first-person mode. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 68

5.4 The visibility tool helps to analyze line-of-sight from a chosen position. **a)** The sphere is placed in the corner of a building. Visible areas from the position of the sphere are colored in a green shade. **b)** If the sphere is placed inside a building, nothing outside the building is visible. Image **c)** shows that from the current position of the sphere, people would not see the bottom of the house on the left side. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69
5.5 The desktop user and the immersed user can use several tools to retrieve the quantitative properties of scene objects. Image **a)-b)** show the retrieval of quantitative information from the desktop and image **c)-d)** depict a white ruler, the measurement tool provided for the immersed user. . . . . . . . . . 70
5.6 Terrain test data obtained from the IMG. **a)-c) and e)** are based on DEM models and **d) and f)** are reconstructed using aerial imaging. . . . . . . . 73
5.7 Occlusion is a common problem when visualizing different layers of information at once. Due to the topology of the terrain data, part of a line, shown in white, is occluded. **Left:** From the perspective of the desktop user and. **Right:** From the perspective of the immersed user. . . . . . . . . . . . . . . . . . . 75
5.8 The desktop user can adjust the position of a desktop marker to avoid occlusion. **a)-b)** Marker elements do not rotate with the map view. A white line indicates the exact location of a marker. **c)** The terrain texture can be disabled, to reduce visual clutter. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75

# List of Tables

2.1 Design considerations for effective manipulation techniques in virtual spaces proposed by Pinho et al. [PBF02] . . . . . . . . . . . . . . . . . . . . . . . . 17

3.1 Overview of the user roles and their objectives depending on the phase of the staff exercise . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31

5.1 List of selected terrain test data files, their properties and import times. . 73

# Acronyms

**AR** Augmented Reality. 1, 10, 15, 16, 20, 29

**CAVE** Cave Automatic Virtual Environment. 9, 10, 79

**CPU** Central Processing Unit. 39, 41, 73

**CVE** Collaborative Virtual Environment. 13–15, 17, 18, 21

**DEM** Digital Elevation Model. 27, 43, 73, 80, 82

**DoF** Degrees-of-Freedom. 55

**FPS** Frames Per Second. 73, 74

**GIS** Geographic Information System. ix, 2, 3, 7–10, 20, 46, 72, 79

**GPU** Graphics Processing Unit. 39, 52, 53, 73

**GUI** Graphical User Interface. 31, 40, 48

**HCI** Human Computer Interaction. 11, 12

**HMD** Head-Mounted Display. ix, 1–3, 9–11, 15, 26, 29, 79

**IDE** Integrated Development Environment. 40

**IMG** Institute for Military Geography. vii, ix, xi, 10, 43, 65, 73, 74, 77, 82

**IPD** Interpupillary Distance. 30

**JSON** JavaScript Object Notation. 44, 45

**NATO** North Atlantic Treaty Organization. 19

**OBJ** Wavefront 3D File. 43

**OOP** Object Oriented Programming. 40

**RAM** Random-Access Memory. 73

**SDK** Software Development Kit. 39

**UI** User Interface. 13, 29, 31, 32, 38, 46, 48, 49, 56, 80

**UTM** Universal Transverse Mercator. 27, 45–47, 49, 53, 58, 59, 62

**VE** Virtual Environment. 8, 18

**VGE** Virtual Geographical Environment. ix, xi, 8–10, 25, 28

**VR** Virtual Reality. ix, xi, 1–4, 9–13, 15, 16, 18–20, 26–33, 35, 37, 39, 41, 42, 45–50, 53, 55–63, 65–81

**VRGIS** Virtual Reality Geographical Information Systems. 2, 9, 10

**VRML** Virtual Reality Modeling Language. 2

**VST** Video-See-Through. 29

**WGS** World Geodetic System. 7, 47

**WIM** World-In-Miniature. 34, 57

**WKT** Well-known Text. 46

**WYSIWYG** What You See Is What You Get. 40

# Bibliography


[ACY18] Akhan Akbulut, Cagatay Catal, and Burak Yıldız. On the effectiveness of virtual reality in the education of software engineering. *Computer Applications in Engineering Education*, 26(4):918–927, 2018.

[AG00] Thomas Alexander and Klaus-Peter Gaertner. The electronic sandtable: An application of VE-technology as tactical situation display. Technical report, Research Establishment for Applied Sciences Wachtberg-Werthhoven (Germany), 2000.

[AG17] Sonam Agrawal and Rakesh Gupta. Web GIS and its architecture: a review. *Arabian Journal of Geosciences*, 10(23):1–13, 2017.

[AGR17] Mahdi Azmandian, Timofey Grechkin, and Evan Suma Rosenberg. An evaluation of strategies for two-user redirected walking in shared physical spaces. In *2017 IEEE Virtual Reality(VR)*, pages 91–98. IEEE, 2017.

[ARD+19] Stephen A Alexander, Juan S Rozo, Bianca T Donadio, Nathan L Tenhundfeld, Ewart J de Visser, and Chad C Tossell. Transforming the air force mission planning process with virtual and augmented reality. In *2019 Systems and Information Engineering Design Symposium (SIEDS)*, pages 1–4. IEEE, 2019.

[Aut21] Autodesk. Overview AutoCAD, 3D map tools. `https://www.autodesk.com/products/autocad/included-toolsets/autocad-map-3d`, 2021. accessed on: 2021-07-12.

[BDHB99] Doug A Bowman, Elizabeth T Davis, Larry F Hodges, and Albert N Badre. Maintaining spatial orientation during travel in an immersive virtual environment. *Presence*, 8(6):618–631, 1999.

[BGR+18] Philippe Bertrand, Jérôme Guegan, Léonore Robieux, Cade Andrew McCall, and Franck Zenasni. Learning empathy through virtual reality: multiple strategies for training empathy-related abilities using body ownership illusions in embodied virtual reality. *Frontiers in Robotics and AI*, 5:26, 2018.

[BGRP01] Steve Benford, Chris Greenhalgh, Tom Rodden, and James Pycock. Collaborative virtual environments. *Communications of the ACM*, 44(7):79–85, 2001.

[BH16] Axel Bédé and Gesine Hofinger. Stabsrahmenübungen für krisenstäbe. In *Handbuch Stabsarbeit*, pages 243–249. Springer, 2016.

[BJH01] Doug A Bowman, Donald B Johnson, and Larry F Hodges. Testbed evaluation of virtual environment interaction techniques. *Presence*, 10(1):75–95, 2001.

[BKS99] Martin Bauer, Gerd Kortuem, and Zary Segall. " where are you pointing at?" a study of remote collaboration in a wearable videoconference system. In *Digest of Papers. Third International Symposium on Wearable Computers*, pages 151–158. IEEE, 1999.

[BM07] D. A. Bowman and R. P. McMahan. Virtual reality: How much immersion is enough? *Computer*, 40(7):36–43, 2007.

[BPP95] Gavin Bell, Anthony Parisi, and Mark Pesce. The virtual reality modeling language. *vol. Version*, 1, 1995.

[BQW20] Ghazi Balkiz, Tamara Qiblawi, and Ben Wedeman. Beirut explosion shatters windows across lebanese capital. `https://web.archive.org/web/20200804212749/https://www.cnn.com/2020/08/04/middleeast/beirut-explosion-port-intl/index.html`, 2020. accessed on: 2021-09-25.

[BRS+19] Michael W. Boyce, Charles P. Rowan, Paul L. Shorter, Jason D. Moss, Charles R. Amburn, Christopher J. Garneau, and Robert A. Sottilare. The impact of surface projection on military tactics comprehension. *Military Psychology*, 31(1):45–59, 2019.

[BSC+18] Miguel Borges, Andrew Symington, Brian Coltin, Trey Smith, and Rodrigo Ventura. Htc vive: Analysis and accuracy improvement. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2610–2615. IEEE, 2018.

[Bul12] Saniye Tugba Bulu. Place presence, social presence, co-presence, and satisfaction in virtual worlds. *Computers & Education*, 58(1):154–161, 2012.

[BV17] Leif P Berg and Judy M Vance. Industry use of virtual reality in product design and manufacturing: a survey. *Virtual reality*, 21(1):1–17, 2017.

[BYB+08] Jeremy N Bailenson, Nick Yee, Jim Blascovich, Andrew C Beall, Nicole Lundblad, and Michael Jin. The use of immersive virtual reality in the learning sciences: Digital transformations of teachers, students, and social context. *The Journal of the Learning Sciences*, 17(1):102–141, 2008.

[CB61] C Comeau and J. Bryan. Headsight television system provides remote surveillance. *Electronics*, pages 86–90, 1961.

[CB16] James J Cummings and Jeremy N Bailenson. How immersive is enough? a meta-analysis of the effect of immersive technology on user presence. *Media Psychology*, 19(2):272–309, 2016.

[CKW13] Hung-Lin Chi, Shih-Chung Kang, and Xiangyu Wang. Research trends and opportunities of augmented reality applications in architecture, engineering, and construction. *Automation in construction*, 33:116–122, 2013.

[CKY20] Eunhee Chang, Hyun Taek Kim, and Byounghyun Yoo. Virtual reality sickness: a review of causes and measurements. *International Journal of Human–Computer Interaction*, 36(17):1658–1682, 2020.

[CL18] Min Chen and Hui Lin. Virtual geographic environments (VGEs): originating from or beyond virtual reality (VR)?, 2018.

[CLL17] Min Chen, Hui Lin, and Guonian Lu. *Virtual Geographic Environments*, pages 1–11. American Cancer Society, 2017.

[CMH16] Philippe A Cerfontaine, Anne-Sophie Mreyen, and Hans-Balder Havenith. Immersive visualization of geophysical data. In *2016 International Conference on 3D Imaging (IC3D)*, pages 1–6. IEEE, 2016.

[CNSD+92] Carolina Cruz-Neira, Daniel J Sandin, Thomas A DeFanti, Robert V Kenyon, and John C Hart. The cave: audio visual experience automatic virtual environment. *Communications of the ACM*, 35(6):64–73, 1992.

[Cor21a] HTC Corporation. Vive pro - premium VR for hardcore gaming. `https://www.vive.com/eu/product/vive-pro/`, 2021. accessed on: 2021-08-01.

[Cor21b] Valve Corporation. SteamVR unity plugin. `https://valvesoftware.github.io/steamvr_unity_plugin/`, 2021. accessed on: 2021-09-02.

[CS98] Elizabeth Churchill and Dave Snowdon. Collaborative virtual environments: an introductory review of issues and systems. *virtual reality*, 3(1):3–15, 1998.

[DF09] Thierry Duval and Cédric Fleury. An asymmetric 2D pointer/3D ray for 3D interaction within collaborative virtual environments. In *Proceedings of the 14th international Conference on 3D Web Technology*, pages 33–41, 2009.

[DH07] Raimund Dachselt and Anett Hübner. Three-dimensional menus: A survey and taxonomy. *Computers & Graphics*, 31(1):53–65, 2007.

[DYLM20] Weihua Dong, Tianyu Yang, Hua Liao, and Liqiu Meng. How does map use differ in virtual reality and desktop-based environments? *International Journal of Digital Earth*, 13(12):1484–1503, 2020.

[DZSZ18] Jing Du, Zhengbo Zou, Yangming Shi, and Dong Zhao. Zero latency: Real-time synchronization of bim data in virtual reality for collaborative decision-making. *Automation in Construction*, 85:51–64, 2018.

[Ear21] About Google Earth. Google earth. `https://earth.google.com/web`, 2021. accessed on: 2021-07-12.

[EET93] Hossam ElGindy, Hazel Everett, and Godfried Toussaint. Slicing an ear using prune-and-search. *Pattern Recognition Letters*, 14(9):719–722, 1993.

[EJAL16] Mariele Evers, Andreja Jonoski, Adrian Almoradie, and Leonie Lange. Collaborative decision making in sustainable flood risk management: A socio-technical approach and tools for participatory governance. *Environmental Science & Policy*, 55:335–344, 2016.

[ELL+18] Carmine Elvezio, Frank Ling, Jen-Shuo Liu, Barbara Tversky, and Steven Feiner. Collaborative exploration of urban data in virtual and augmented reality. In *ACM SIGGRAPH 2018 Virtual, Augmented, and Mixed Reality*, SIGGRAPH '18, New York, NY, USA, 2018. Association for Computing Machinery.

[EMM20] Paul MG Emmelkamp, Katharina Meyerbröker, and Nexhmedin Morina. Virtual reality therapy in social anxiety disorder. *Current psychiatry reports*, 22(7), 2020.

[ESFT17] Carmine Elvezio, Mengu Sukan, Steven Feiner, and Barbara Tversky. Travel in large-scale head-worn VR: Pre-oriented teleportation with wims and previews. In *2017 IEEE Virtual Reality(VR)*, pages 475–476. IEEE, 2017.

[ESO+17] Carmine Elvezio, Mengu Sukan, Ohan Oda, Steven Feiner, and Barbara Tversky. Remote collaboration in AR and VR using virtual replicas. In *ACM SIGGRAPH 2017 VR Village*, SIGGRAPH '17, New York, NY, USA, 2017. Association for Computing Machinery.

[Esr21] Esri. About ArcGIS. `https://www.esri.com/en-us/arcgis/3d-gis/overview`, 2021. accessed on: 2021-07-12.

[fLB21] Bundesministerium für Landesverteidigung (BMLV). Geschichte der theresianische militärakademie. `https://www.milak.at/`

informationen-fuer-besucher/burg#event-169-21, 2021. accessed on: 2021-09-12.

[FMR+18] Cameron G Ford, Ellen M Manegold, Cameron L Randall, Ariel M Aballay, and Christina L Duncan. Assessing the feasibility of implementing low-cost virtual reality therapy during routine burn care. *Burns*, 44(4):886–895, 2018.

[Fou21] Blender Foundation. The official blender foundation tree. https://git.blender.org/gitweb/gitweb.cgi/blender.git, 2021. accessed on: 2021-09-02.

[FT21] LLC. Facebook Technologies. Oculus quest 2 - all-in-one. one for all. https://www.oculus.com/quest-2/, 2021. accessed on: 2021-08-01.

[GB95] Chris Greenhalgh and Steven Benford. Massive: A collaborative virtual environment for teleconferencing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 2(3):239–261, 1995.

[GCM+19] Scott W Greenwald, Wiley Corning, Gavin McDowell, Pattie Maes, and John Belcher. Electrovr: An electrostatic playground for collaborative, simulation-based exploratory learning in immersive virtual reality. In Kristine Lund, Gerald P. Niccolai, Élise Lavoué, Cindy E. Hmelo-Silver, Gahgene Gweon, and Michael Baker, editors, *13th International Conference on Computer Supported Collaborative Learning, CSCL 2019, Lyon, France, June 17-21, 2019*. International Society of the Learning Sciences, 2019.

[GDM19] Jerônimo Gustavo Grandi, Henrique Galvan Debarba, and Anderson Maciel. Characterizing asymmetric collaborative interactions in virtual and augmented realities. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces(VR)*, pages 127–135. IEEE, 2019.

[GFDC20] Raluca Georgescu, Liviu A. Fodor, Anca Dobrean, and Ioana A. Cristea. Psychological interventions using virtual reality for pain associated with medical procedures: a systematic review and meta-analysis. *Psychological Medicine*, 50(11):1795–1807, 2020.

[GFR+19] Arturo García, T. Fernando, David Roberts, Christian Bar, Michele Cencetti, Wito Engelke, and Andreas Gerndt. Collaborative virtual reality platform for visualizing space data and mission planning. *Multimedia Tools and Applications*, 78, 12 2019.

[GKK+17] Scott W Greenwald, Alexander Kulik, André Kunert, Stephan Beck, Bernd Fröhlich, Sue Cobb, Sarah Parsons, Nigel Newbutt, Christine Gouveia, Claire Cook, et al. Technology and applications for collaborative learning in virtual reality. In *CSCL*. Philadelphia, PA: International Society of the Learning Sciences., 2017.

[Gmb21] Wikitude GmbH. Wikitude GmbH - create cutting-edge AR apps. `https://www.wikitude.com/`, 2021. accessed on: 2021-08-01.

[GMMG08] Arturo S García, José P Molina, Diego Martínez, and Pascual González. Enhancing collaborative manipulation through the use of feedback and awareness in cves. In *Proceedings of the 7th ACM SIGGRAPH international Conference on Virtual-Reality Continuum and Its Applications in industry*, pages 1–5, 2008.

[Goo05] Michael F Goodchild. GIS and modeling overview. *GIS, spatial analysis, and modeling. ESRI Press, Redlands*, pages 1–18, 2005.

[Goo12] Michael F. Goodchild. Geographic information systems. In Alan Hastings and Louis Gross, editors, *Encyclopedia of Theoretical Ecology*, pages 341–345. University of California Press, 2012.

[Gro21] Khronos Group. OpenXR - a royalty-free, open standard for augmented reality (AR) and virtual reality(VR). `https://www.khronos.org/openxr/`, 2021. accessed on: 2021-08-10.

[GSFR17] Jan Gugenheimer, Evgeny Stemasov, Julian Frommel, and Enrico Rukzio. Sharevr: Enabling co-located experiences for virtual reality between hmd and non-hmd users. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4021–4033, 2017.

[gvS21] gvSIG. gvsig desktop, the open source geographic information system. `http://www.gvsig.org/en/web/guest/products`, 2021. accessed on: 2021-07-12.

[Hak02] Mordechay E Haklay. Virtual reality and GIS: Applications. trends and directions. In *In Fisher, P. & Unwin, D.(eds), Virtual Reality in Geography*. Citeseer, 2002.

[HAM+03] Jeffrey Hogue, R Allen, Steve Markham, Arvid Harmsen, Jerry MacDonald, and Cliff Schmucker. Parachute mission planning, training and rehearsal using a deployable virtual reality simulator. In *17th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, page 2156, 2003.

[HCM19] Hans-Balder Havenith, Philippe Cerfontaine, and Anne-Sophie Mreyen. How virtual reality can help visualise and assess geohazards. *International Journal of Digital Earth*, 12(2):173–189, 2019.

[HCR+19] Kelly S Hale, Gwen Campbell, Jennifer Riley, Michael Boyce, and Charles Amburn. Augmented reality sandtable (ARES) impacts on learning. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 63(1):2149–2153, 2019.

[HCW88] Don Hopkins, Jack Callahan, and Mark Weiser. Pies: implementation, evaluation and application of circular menus. *University of Maryland Computer Science Department Technical Report*, 1988.

[Hei60] Morton L Heilig. Stereoscopic-television apparatus for individual use, 10 1960. US Patent 2,955,156.

[HG20] Matt C Howard and Melissa B Gutworth. A meta-analysis of virtual reality training programs for social skill development. *Computers & Education*, 144:103707, 2020.

[HH07] Khaled Hassanein and Milena Head. Manipulating perceived social presence through the web interface and its impact on attitude towards online shopping. *International journal of human-computer studies*, 65(8):689–708, 2007.

[HH16] Rudi Heimann and Gesine Hofinger. Stabsarbeit–konzept und formen der umsetzung. In *Handbuch Stabsarbeit*, pages 3–9. Springer, 2016.

[Hil12] Sébastien Hillaire. Antialiased volumetric lines using shader-based extrusion. In Patrick Cozzi and Christophe Riccio, editors, *OpenGL insights*, chapter 11, pages 163–171. CRC Press, 2012.

[Hol21] Freya Holmér. Shapes. `https://acegikmo.com/shapes/`, 2021. accessed on: 2021-09-02.

[HRL14] Erich Heumüller, Sebastian Richter, and Ulrike Lechner. Ein konzeptionelles stabsmodell für die entwicklung von stabsrahmenübungen. *Proceedings der Multikonferenz Wirtschaftsinformatik (MKWI 2014)*, pages 321–334, 2014.

[HRP19] Zhenyi He, Karl Toby Rosenberg, and Ken Perlin. Exploring configuration of mixed reality spaces for communication. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2019.

[HŠM+19] Nikola Horvat, Stanko Škec, Tomislav Martinec, Fanika Lukačević, and Marija Majda Perišić. Comparing virtual reality and desktop interface for reviewing 3D cad models. *Proceedings of the Design Society: International Conference on Engineering Design*, 1(1):1923–1932, 2019.

[iGEV21] Source images Google Earth VR. Official steam store google earth VR. `https://store.steampowered.com/app/348250/Google_Earth_VR/`, 2021. accessed on: 2021-07-25.

[Jer15] Jason Jerald. *The VR book: Human-centered design for virtual reality.* Morgan & Claypool, 2015.

[KD10] Jan Klimke and Jürgen Döllner. Combining synchronous and asynchronous collaboration within 3D city models. In *International Conference on Geographic Information Science*, pages 115–129. Springer, 2010.

[KFS03] Robert E Kraut, Susan R Fussell, and Jane Siegel. Visual information as a conversational resource in collaborative physical tasks. *Human–computer interaction*, 18(1-2):13–49, 2003.

[KGS12] Konstantina Kilteni, Raphaela Groten, and Mel Slater. The sense of embodiment in virtual reality. *Presence: Teleoperators and Virtual Environments*, 21(4):373–387, 2012.

[KLBL93] Robert S Kennedy, Norman E Lane, Kevin S Berbaum, and Michael G Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology*, 3(3):203–220, 1993.

[KLH+19] Seungwon Kim, Gun Lee, Weidong Huang, Hayun Kim, Woontack Woo, and Mark Billinghurst. Evaluating the combination of visual communication cues for hmd-based mixed reality remote collaboration. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–13, 2019.

[KPG+17] Dominik P. Käser, Evan Parker, Adam Glazier, Mike Podwal, Matt Seegmiller, Chun-Po Wang, Per Karlsson, Nadav Ashkenazi, Joanna Kim, Andre Le, Matthias Bühlmann, and Joshua Moshier. The making of google earth VR. In *ACM SIGGRAPH 2017 Talks*, SIGGRAPH '17, New York, NY, USA, 2017. Association for Computing Machinery.

[Kru12] Ernst Kruijff. 3D user interfaces: theory and practice. *IEEE transactions on visualization and computer graphics*, 18(4):565–572, 2012.

[KSD00] Robert S Kennedy, Kay M Stanney, and William P Dunlap. Duration and exposure to virtual environments: sickness curves during and across sessions. *Presence: Teleoperators & Virtual Environments*, 9(5):463–472, 2000.

[KSD+19] Katharina Krösl, Harald Steinlechner, Johanna Donabauer, Daniel Cornel, and Jürgen Waser. Master of disaster. In *The 17th International Conference on Virtual-Reality Continuum and its Applications in Industry*, pages 1–2, 2019.

[KSP+19] Bhone Myint Kyaw, Nakul Saxena, Pawel Posadzki, Jitka Vseteckova, Charoula Konstantia Nikolaou, Pradeep Paul George, Ushashree Divakar, Italo Masiello, Andrzej A Kononowicz, Nabil Zary, et al. Virtual reality for health professions education: systematic review and meta-analysis by the digital health education collaboration. *Journal of medical Internet research*, 21(1):e12959, 2019.

[Kum88] Muneendra Kumar. World geodetic system 1984: A modern and accurate global reference frame. *Marine Geodesy*, 12(2):117–126, 1988.

[KWS94] Jinchoon Kim, Richard Weidner, and Allan Sacks. Using virtual reality for science mission planning: A mars pathfinder case. *NASA. Johnson Space Center, ISMCR 1994: Topical Workshop on Virtual Reality. Proceedings of the Fourth International Symposium on Measurement and Control in Robotics*, 1994.

[Lan98] Richard B Langley. The UTM grid system. *GPS world*, 9(2):46–50, 1998.

[Lel13] Ajey Lele. Virtual reality and its military utility. *Journal of Ambient Intelligence and Humanized Computing*, 4(1):17–26, 2013.

[LJKM+17] Joseph J LaViola Jr, Ernst Kruijff, Ryan P McMahan, Doug Bowman, and Ivan P Poupyrev. *3D user interfaces: theory and practice*. Addison-Wesley Professional, 2017.

[LPBR17] Jérémy Lacoche, Nico Pallamin, Thomas Boggini, and Jérôme Royan. Collaborators awareness for user cohabitation in co-located collaborative virtual environfments. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, pages 1–9, 2017.

[LVW+21] Jie Li, Vinoba Vinayagamoorthy, Julie Williamson, David A Shamma, and Pablo Cesar. Social VR: A new medium for remote communication and collaboration. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2021.

[Map21] Mapbox. About mapbox. `https://www.mapbox.com/mapbox-studio/`, 2021. accessed on: 2021-07-12.

[MBGM19] Guido Makransky, Stefan Borre-Gude, and Richard E Mayer. Motivational and cognitive benefits of training in immersive virtual reality based on multiple assessments. *Journal of Computer Assisted Learning*, 35(6):691–707, 2019.

[Mer96] Daniel F Merriam. Kansas 19th century geologic maps. *Transactions of the Kansas Academy of Science (1903)*, pages 95–114, 1996.

[MG17] Kapil Chalil Madathil and Joel S Greenstein. An investigation of the efficacy of collaborative virtual reality systems for moderated remote usability testing. *Applied Ergonomics*, 65:501–514, 2017.

[MGDB+19] Marta Matamala-Gomez, Tony Donegan, Sara Bottiroli, Giorgio Sandrini, Maria V Sanchez-Vives, and Cristina Tassorelli. Immersive virtual reality and virtual embodiment for pain relief. *Frontiers in human neuroscience*, 13:279, 2019.

[MGFS13] Mikhail Morozov, Alexey Gerasimov, Mikhail Fominykh, and Andrey Smorkalov. Asynchronous immersive classes in a 3D virtual world: Extended description of vacademia. In *Transactions on computational science XVIII*, pages 81–100. Springer, 2013.

[Mic21a] Microsoft. MRTK-Unity3D is a microsoft-driven project. `https://github.com/microsoft/MixedRealityToolkit-Unity`, 2021. accessed on: 2021-08-01.

[Mic21b] Microsoft. The place for events. `https://altvr.com`, 2021. accessed on: 2021-08-01.

[MK16] Tobias Meier and Norbert Kanschus. Mobile und abgesetzte befehlsstellen am beispiel eines spezialeinsatzkommandos. In *Handbuch Stabsarbeit*, pages 145–152. Springer, 2016.

[MMB20] Ali Ahmad Malik, Tariq Masood, and Arne Bilberg. Virtual reality in manufacturing: immersive and collaborative artificial-reality in design of human-robot workspace. *International Journal of Computer Integrated Manufacturing*, 33(1):22–37, 2020.

[MOM19] Oliver A. Meyer, Magnus K. Omdahl, and Guido Makransky. Investigating the effect of pre-training when learning through immersive virtual reality and video: A media and methods experiment. *Computers & Education*, 140:103603, 2019.

[MSMSMI18] Joshua McVeigh-Schultz, Elena Márquez Segura, Nick Merrill, and Katherine Isbister. What's it mean to" be social" in VR? mapping the social VR design ecology. In *Proceedings of the 2018 ACM Conference Companion Publication on Designing Interactive Systems*, pages 289–294, 2018.

[MWA+19] Carl Matthes, Tim Weissker, Emmanouil Angelidis, Alexander Kulik, Stephan Beck, Andre Kunert, Anton Frolov, Sandro Weber, Adrian Kreskowski, and Bernd Froehlich. The collaborative virtual reality neurorobotics lab. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces(VR)*, pages 1671–1674. IEEE, 2019.

[MYW+12] N Menck, X Yang, C Weidig, P Winkes, C Lauer, H Hagen, B Hamann, and JC Aurich. Collaborative factory planning in virtual reality. *Procedia CIRP*, 3:317–322, 2012.

[NAT21] NATO. NATO joint military symbology, app-6(c). https://web.archive.org/web/20150921231042/http://armawiki.zumorc.de/files/NATO/APP-6(C).pdf, 2021. accessed on: 2021-09-05.

[ND14] Thi Thuong Huyen Nguyen and Thierry Duval. A survey of communication and awareness in collaborative virtual environments. In *2014 International Workshop on Collaborative Virtual Environments (3DCVE)*, pages 1–8. IEEE, 2014.

[NLSB19] Mitchell Norman, Gun A Lee, Ross T Smith, and Mark Billingurst. The impact of remote user's role in a mixed reality mixed presence system. In *The 17th International Conference on Virtual-Reality Continuum and its Applications in Industry*, pages 1–9, 2019.

[NM90] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256, 1990.

[OBW18] Catherine S Oh, Jeremy N Bailenson, and Gregory F Welch. A systematic review of social presence: Definition, antecedents, and implications. *Frontiers in Robotics and AI*, 5:114, 2018.

[Ope21] About OpenStreetMap. OpenStreetMap provides map data for thousands of websites, mobile apps, and hardware devices. https://www.openstreetmap.org/about, 2021. accessed on: 2021-07-12.

[ORW06] Oliver Otto, Dave Roberts, and Robin Wolff. A review on effective closely-coupled collaboration using immersive cve's. In *Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*, pages 145–154, 2006.

[OS93] Judith Orasanu and Eduardo Salas. Team decision making in complex environments. In Jan Fagerberg, David C. Mowery, and Richard R. Nelson, editors, *Decision making in action: Models and methods.*, pages 327–345. Ablex Publishing, 1993.

[PA20] Catlin Pidel and Philipp Ackermann. Collaboration in virtual and augmented reality: a systematic overview. In *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, pages 141–156. Springer, 2020.

[PBF02] Márcio S Pinho, Doug A Bowman, and Carla MDS Freitas. Cooperative object manipulation in immersive virtual environments: framework and techniques. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 171–178, 2002.

[PDE+19] Thammathip Piumsomboon, Arindam Dey, Barrett Ens, Gun Lee, and Mark Billinghurst. The effects of sharing awareness cues in collaborative mixed reality. *Frontiers in Robotics and AI*, 6:5, 2019.

[PT89] Herbert L Pick and William B Thompson. Topographic map reading. Technical report, Minnesota University Minneapolis, 1989.

[PTC21] PTC. Vuforia: Market-leading enterprise AR. `https://www.ptc.com/en/products/vuforia`, 2021. accessed on: 2021-08-01.

[PVS+16] Iana Podkosova, Khrystyna Vasylevska, Christian Schoenauer, Emanuel Vonach, Peter Fikar, Elisabeth Bronederk, and Hannes Kaufmann. Immersivedeck: A large-scale wireless VR system for multiple users. In *2016 IEEE 9th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*, pages 1–7. IEEE, 2016.

[PWB+15] Andrea Philips, Ariane Walz, Andreas Bergner, Thomas Graeff, Maik Heistermann, Sarah Kienzler, Oliver Korup, Torsten Lipp, Wolfgang Schwanghart, and Gerold Zeilinger. Immersive 3D geovisualization in higher education. *Journal of Geography in Higher Education*, 39(3):437–449, 2015.

[QG16] Andreas Queck and Harald Gonner. Informationsmanagement im krisenstab. In *Handbuch Stabsarbeit*, pages 183–190. Springer, 2016.

[QGI21] QGIS. A free and open source geographic information system. `https://www.qgis.org/en/site/`, 2021. accessed on: 2021-07-12.

[Rei21] Ricardo Reis. Trilib is a cross-platform runtime 3D model importer for Unity3D. `https://ricardoreis.net/trilib-2/`, 2021. accessed on: 2021-09-02.

[RF03] NATO Research and Technology Organization Neuilly-Sur-Seine (France). Virtual reality: State of military research and applications in member countries. *RTO Technical Report 18*, 2003.

[RHWF06] Kai Riege, Thorsten Holtkamper, Gerold Wesche, and Bernd Frohlich. The bent pick ray: An extended pointing technique for multi-user interaction. In *3D User Interfaces (3DUI'06)*, pages 62–65. IEEE, 2006.

[RLN07] Ruth Rosenholtz, Yuanzhen Li, and Lisa Nakano. Measuring visual clutter. *Journal of vision*, 7(2):17–17, 2007.

[RMFW20] Jaziar Radianti, Tim A Majchrzak, Jennifer Fromm, and Isabell Wohlgenannt. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & Education*, 147:103778, 2020.

[Rou13] Mattias Roupé. *Development and implementations of virtual reality for decision-making in urban planning and building design*. Chalmers Tekniska Hogskola (Sweden), 2013.

[RSMP11] Che Mat Ruzínoor, Abdul Rashid Mohamed Shariff, Ahmad Rodzi Mahmud, and Biswajeet Pradhan. 3D terrain visualisation for GIS: A comparison of different techniques. In *True-3D in Cartography*, pages 265–277. Springer, 2011.

[RWM19] Giuseppe Riva, Brenda K Wiederhold, and Fabrizia Mantovani. Neuroscience of virtual reality: from virtual exposure to embodied medicine. *Cyberpsychology, Behavior, and Social Networking*, 22(1):82–96, 2019.

[RWOS03] David Roberts, Robin Wolff, Oliver Otto, and Anthony Steed. Constructing a gazebo: supporting teamwork in a tightly coupled, distributed task in virtual reality. *Presence*, 12(6):644–657, 2003.

[SAT21] Anthony Scavarelli, Ali Arya, and Robert J Teather. Virtual reality and augmented reality in social learning spaces: A literature review. *Virtual Reality*, 25:257–277, 2021.

[SAX+09] Konstantinos Smparounis, Kosmas Alexopoulos, Vagelis Xanthakis, Menelaos Pappas, Dimitris Mavrikios, and George Chryssolouris. A web-based platform for collaborative product design and evaluation. In *2009 IEEE International Technology Management Conference (ICE)*, pages 1–9. IEEE, 2009.

[SB07] Philip Schuchardt and Doug A Bowman. The benefits of immersion for spatial understanding of complex underground cave systems. In *Proceedings of the 2007 ACM symposium on Virtual reality software and technology*, pages 121–124, 2007.

[SB09a] Stefan Steiniger and Erwan Bocher. An overview on current free and open source desktop GIS developments. *International Journal of Geographical Information Science*, 23(10):1345–1370, 2009.

[SB09b] Stefan Steiniger and Erwan Bocher. An overview on current free and open source desktop GIS developments. *International Journal of Geographical Information Science*, 23(10):1345–1370, 2009.

[SCG21] Based in Canada Steel Crate Games. The co-op bomb defusing party game. `https://keeptalkinggame.com`, 2021. accessed on: 2021-07-31.

[SCM01] Dave Snowdon, Elizabeth F Churchill, and Alan J Munro. Collaborative virtual environments: Digital spaces and places for cscw: An introduction. In *Collaborative virtual environments*, pages 3–17. Springer, 2001.

[SCP95] Richard Stoakley, Matthew J Conway, and Randy Pausch. Virtual reality on a wim: interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 265–272, 1995.

[SD19] Yusuf Sermet and Ibrahim Demir. Flood action VR: A virtual reality framework for disaster awareness and emergency response training. In *ACM SIGGRAPH 2019 Posters*, SIGGRAPH '19, New York, NY, USA, 2019. Association for Computing Machinery.

[SDRH+16] Tarah N Schmidt-Daly, Jennifer M Riley, Kelly S Hale, David Yacht, and Jack Hart. Augmented reality sandtables (ARESs) impact on learning. Technical report, Design Interactive, Inc. Orlando United States, 2016.

[SGBB19] Prasanth Sasikumar, Lei Gao, Huidong Bai, and Mark Billinghurst. Wearable remotefusion: A mixed reality remote collaboration system with local eye gaze and remote hand gesture sharing. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 393–394. IEEE, 2019.

[Sha20] Sharad Sharma. Improving emergency response training and decision making using a collaborative virtual reality environment for building evacuation. In *International Conference on Human-Computer Interaction*, pages 213–224. Springer, 2020.

[SIC+18] Yuta Sugiura, Hikaru Ibayashi, Toby Chong, Daisuke Sakamoto, Natsuki Miyata, Mitsunori Tada, Takashi Okuma, Takeshi Kurata, Takashi Shinmura, Masaaki Mochimaru, et al. An asymmetric collaborative system for architectural-scale space design. In *Proceedings of the 16th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, pages 1–6, 2018.

[SK03] Marc A Schnabel and Thomas Kvan. Spatial understanding in immersive virtual environments. *International Journal of Architectural Computing*, 1(4):435–448, 2003.

[Sla03] Mel Slater. A note on presence terminology. *Presence connect*, 3(3):1–5, 2003.

[SMGG01] Peter-Pike J Sloan, William Martin, Amy Gooch, and Bruce Gooch. The lit sphere: A model for capturing npr shading from art. In *Graphics interface*, volume 2001, pages 143–150. Citeseer, 2001.

[SMK+09] Gerhard Schall, Erick Mendez, Ernst Kruijff, Eduardo Veas, Sebastian Junghanns, Bernhard Reitinger, and Dieter Schmalstieg. Handheld augmented reality for underground infrastructure visualization. *Personal and ubiquitous computing*, 13(4):281–291, 2009.

[SMR+18] Jonas Schild, Sebastian Misztal, Beniamin Roth, Leonard Flock, Thomas Luiz, Dieter Lerner, Markus Herkersdorf, Konstantin Weaner, Markus Neuberaer, Andreas Franke, et al. Applying multi-user virtual reality to collaborative medical training. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces(VR)*, pages 775–776. IEEE, 2018.

[Sof21] Valve Software. Openvr sdk. `https://github.com/ValveSoftware/openvr`, 2021. accessed on: 2021-09-1.

[SPB+20] Simon Su, Vince Perry, Luis Bravo, Sue Kase, Heather Roy, Katherine Cox, and Venkat R Dasari. Virtual and augmented reality applications to support data analysis and assessment of science and engineering. *Computing in Science & Engineering*, 22(3):27–39, 2020.

[SRD15] Sharad Sharma, Shanmukha Pranay Rajeev, and Phillip Devearux. An immersive collaborative virtual environment of a university campus for performing virtual campus evacuation drills and tours for campus safety. In *2015 International Conference on Collaboration Technologies and Systems (CTS)*, pages 84–89. IEEE, 2015.

[SS15] Anthony Steed and Ralph Schroeder. Collaboration in immersive and non-immersive virtual environments. In *Immersed in Media*, pages 263–282. Springer, 2015.

[SS21] Inc. Spatial Systems. Spatial - virtual spaces that bring us together. `https://spatial.io`, 2021. accessed on: 2021-08-01.

[SSB+20] Dimitrios Saredakis, Ancret Szpak, Brandon Birckhead, Hannah AD Keage, Albert Rizzo, and Tobias Loetscher. Factors associated with virtual reality sickness in head-mounted displays: a systematic review and meta-analysis. *Frontiers in human neuroscience*, 14:96, 2020.

[Sto21] The Stonefox. VRTK - a productive VR toolkit for rapidly building VR solutions in Unity3D. `https://vrtoolkit.readme.io/`, 2021. accessed on: 2021-08-01.

[Sut65] Ivan E. Sutherland. The ultimate display. In *Proceedings of the IFIP Congress*, pages 506–508, 1965.

[SZDA17] Andrés Santos, Telmo Zarraonandia, Paloma Díaz, and Ignacio Aedo. A comparative study of menus in virtual reality environments. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, pages 294–299, 2017.

[TBG05] Anthony Tang, Michael Boyle, and Saul Greenberg. Understanding and mitigating display and presence disparity in mixed presence groupware. *Journal of research and practice in information technology*, 37(2):193–210, 2005.

[Tec21] Unity Technologies. The leading platform for creating interactive, real-time content. https://unity.com/, 2021. accessed on: 2021-08-022.

[Tho05] C Thome. Using a floating origin to improve fidelity and performance of large, distributed virtual worlds. In *2005 International Conference on Cyberworlds (CW'05)*, pages 8–pp. IEEE, 2005.

[TZS+18] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Facevr: Real-time gaze-aware facial reenactment in virtual reality. *ACM Transactions on Graphics (TOG)*, 37(2):1–15, 2018.

[Unt21] Johannes Unterguggenberger. Volumetric lines. https://assetstore.unity.com/packages/tools/particles-effects/volumetric-lines-29160, 2021. accessed on: 2021-09-10.

[Val21] Valve. Grab SteamVR to access and play VR games using your HTC Vive, Oculus Rift, windows mixed reality headset, or any other supported VR headset and controllers. https://store.steampowered.com/app/250820/SteamVR/, 2021. accessed on: 2021-08-01.

[VR21] Google Earth VR. About google earth VR. https://arvr.google.com/earth/, 2021. accessed on: 2021-07-15.

[VRC21] VRChat Inc. VRChat offers an endless collection of social VR experiences by giving the power of creation to its community. https://hello.vrchat.com, 2021. accessed on: 2021-08-01.

[Wol19] Josef Wolfartsberger. Analyzing the potential of virtual reality for engineering design review. *Automation in Construction*, 104:27–37, 2019.

[WWL+19] Tzong-Hann Wu, Feng Wu, Ci-Jyun Liang, Yi-Fen Li, Ching-Mei Tseng, and Shih-Chung Kang. A virtual reality tool for training in global engineering collaboration. *Universal Access in the Information Society*, 18(2):243–255, 2019.

[Zha03] Shanyang Zhao. Toward a taxonomy of copresence. *Presence*, 12(5):445–455, 2003.

[ZHHL16] Xiaolei Zhang, Yong Han, DongSheng Hao, and Zhihan Lv. ARGIS-based outdoor underground pipeline information system. *Journal of Visual Communication and Image Representation*, 40:779–790, 2016.

[Zim08] Peter Zimmermann. Virtual reality aided design. a survey of the use of VR in automotive industry. In *Product Engineering*, pages 277–296. Springer, 2008.