Hugo Miguel Simões Lino

Bachelor of Science

Collaborative Interaction Techniques in Virtual Reality for Emergency Management

Dissertation submitted in partial fulfillment
of the requirements for the degree of

Master of Science in
Computer Science and Engineering

Adviser: Rui Nóbrega, Assistant Professor, NOVA University of Lisbon

Co-adviser: Fernando Birra, Assistant Professor, NOVA University of Lisbon

Examination Committee

Chair: Sofia Cavaco, Assistant Professor, NOVA University of Lisbon

Rapporteur: Daniel Mendes, Assistant Professor, University of Porto

Members: Rui Nóbrega, Assistant Professor, NOVA University of Lisbon
Fernando Birra, Assistant Professor, NOVA University of Lisbon

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

November, 2021

Collaborative Interaction Techniques in Virtual Reality for Emergency Management

This document was created using the (pdf)LaTeX processor, based on the "novathesis" template[1], developed at the Dep. Informática of FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Acknowledgements

I would like to start by expressing my gratitude to my advisers, Professor Rui Nóbrega and Professor Fernando Birra, for believing in me and guiding me throughout this journey. Without their help this thesis would not be what it is today. I would also like to extend my gratitude to Professor Nuno Correia for helping and reviewing some of my work.

Then, I would like to thank Faculdade de Ciências e Tecnologias da Universidade Nova de Lisboa and all Professors that helped me and taught me over the course of my academic path. I would also like to thank the Fundação para a Ciência e Tecnologia for financing my research through the research grant UIDB/04516/2020/TRA/BIL/15. A thank you should also be extended to the Centro de Informação Geoespacial do Exército for providing some of the high-resolution data used and the Gabinete Florestal de Mação for providing the wildfire data.

A big thank you is owed to my friends and family. A special thanks, in alphabetic order, to; Diogo Cruz, Duarte Rebeca, Gonçalo Rocha, Jorge Fonseca, José Farçadas, Mariana Leite, Pedro Fernandes, Rui Rebeca and Sandra Lino, for helping me and encouraging me, in one way or another, through all my academic years and, especially, in this last step.

A warm thank you to my friend and partner, Inês Messias, for always being there, ready to help, comfort, and listen to my ramblings. You have a special place in my heart.

Last but certainly not least, a very special thank you to my mother, Ana Simões, and my father, Alberto Lino. Without your help, encouragement and love throughout my life I would not be the person I am today.

"Not all complex problems have easy solutions; so says Science
(so warns Science)... "
- Mark Z. Danielewski, House of Leaves

Abstract

Virtual Reality (VR) technology has had many interesting applications over the last decades. It can be seen in a multitude of industries: entertainment, education, tourism to crisis management among others. Many of them, feature collaborative uses of VR technology.

This thesis presents the design, development and evaluation of a multi-user VR system, aimed at collaborative usage focused on a crisis scenario based on real-life wildfire as the use case. The system also features a dual-map interface to display geographical information, providing both two-dimensional and three-dimensional views over the region and data relevant to the scenario. The main goals of this thesis are to understand how people can collaborate in VR, test which interface is preferred, as well as what kinds of notification mechanisms are more user friendly.

The Virtual Environment (VE) displays relevant geo-located information, such as roads, towns, vehicles and the wildfire itself, in a dual-map setup, in two and three dimensions. Users are able to share the environment and, simultaneously, use available tools to interact with the maps and communicate with each other, while controlling the wildfire playback time to understand how it propagates. Actions such as drawing, measuring distances, directing vehicles and notifying other users are available. Users can propose actions that can then be accepted or denied.

Eighteen subjects took part in a user study to evaluate the application. Participants were asked to perform several tasks, using the tools available, while sharing that same environment with the researcher. Upon analyzing data from the testing sessions, it is possible to state that most users agree they would be able to use the system to collaborate. The results also support the presence of both types of map interfaces, two-dimensional and three-dimensional, as they are objectively better suited for different tasks; users, subjectively, affirmed preference for both of them, depending on the task at hand.

Keywords: Virtual Reality, Collaboration, Interaction, Computer Graphics, Crisis Management

Resumo

A Realidade Virtual (RV) tem demonstrado ter várias aplicações interessantes ao longo das últimas décadas. Faz parte de múltiplas indústrias, tais como entertenimento, educação, turismo, gestão de crises, entre outras. Muitas delas usam a tecnologia num contexto colaborativo.

Nesta tese é apresentado o *design*, desenvolvimento e avaliação de um sistema multi-utilizador de RV, dedicado ao uso colaborativo durante um cenário de crise baseado num fogo real. É também implementada uma interface *dual-map* que visualiza informação geográfica, providenciando duas vistas (2D e 3D) sobre a região e dados relevantes ao cenário descrito. Perceber como podem as pessoas colaborar em RV, testar qual a interface preferida e quais os tipos de mecanismos de notificação preferíveis são os objectivos principais desta tese.

O Ambiente Virtual (AV) apresenta informação geo-referenciada relevante, como estradas, povoações, veículos e o próprio incêndio, através da interface dual. Utilizadores podem partilhar o ambiente e, simultaneamente, usar as ferramentas disponíveis para interagir com os mapas e comunicar entre si, enquanto controlam o progresso do incêndio para melhor entender como se propaga. Ações como desenhar, medir distâncias, direcionar veículos e notificar outros utilizadores estão disponíveis. Utilizadores podem também propor ações que serão aceites ou recusadas.

Dezoito pessoas fizeram parte do estudo de utilizador para avaliar a aplicação. Os participantes executaram múltiplas tarefas, usando as ferramentas disponíveis, enquanto partilhavam o mesmo AV que o investigador. Após análise dos dados gerados, é possível afirmar que a maioria dos participantes consideram que seriam capazes de usar o sistema para colaborar. Os resultados também suportam a presença de ambos os tipos de mapas, 2D e 3D, pois ambos são objectivamente melhores para tarefas distintas; participantes, subjectivamente, afirmam preferência por ambas, dependendo da tarefa a executar.

Palavras-chave: Realidade Virtual, Colaboração, Interação, Computação Gráfica, Gestão de Crises

Contents

List of Figures

List of Tables

Acronyms

AR	Augmented Reality
AV	Ambiente Virtual
CRS	Coordinate Reference System
IPQ	Igroup Presence Questionnaire
OSM	Open Street Map
OSRM	Open Source Routing Machine
RPC	Remote Procedure Call
RV	Realidade Virtual
SUS	System Usability Scale
VE	Virtual Environment
VR	Virtual Reality
XR	Extended Reality

Chapter 1

Introduction

Virtual Reality (VR) is a field with immense potential. Being able to transport someone to a place that feels as real as possible can have tremendous impact on that person's experience. It has potential to help with teaching [28], with therapy [4] and it can improve storytelling by placing the user within [15].

VR can also have a significant impact in collaboration. In a GTC 2020 talk[1], Rob Legato, visual effects supervisor, and Ben Grossman, Academy Award-winning virtual production supervisor, detailed the process of remaking the classic Lion King using a complete virtual workflow for filming and acting, all within the Virtual Environment (VE). Another impressive example of collaboration is NVIDIA's Holodeck[2], (in early access since GTC Europe 2017 up to the time of this writing) it allows experts to collaborate and create in a highly realistic VE. Companies such as NASA, Toyota and Koenigsegg have already adopted Holodeck in their design workflow.

It is clear that collaboration is a worthy pursuit in a VR environment. Over the course of this document a collaborative VR system for crisis management will be presented. The system implements collaborative features, interactive tools and interactive visualizations. The application is designed to explore how can a crisis management scenario benefit from being presented in a VE, how users, with and without VR experience, are able to adapt to it and collaborate within it. Another interesting topic to explore is the addition of a dual-map setup, inspired by Medeiros et al. [29] and Coffey et al. [8].

[1] The Lion King: Reinventing the Future of Virtual Production. GTC 2020 talk by Rob Legato and Ben Grossman about how virtual reality impacted the production of the movie. - Last accessed: 28/07/2020

[2] Nvidia Holodeck Virtual reality photorealistic collaborative design tool. - Last accessed: 28/07/2020

1.1 Motivation

Collaboration has always been a needed component of inter-human relations and, within a team, crucial to success. In a distributed world, collaborative tools are a necessity for progress and, as such, are pervasive. Ranging from communications, to planning and visualizations, everyone has had experience with some form of collaboration tool. A new paradigm was introduced when collaborative VR tools, such as the previously mentioned Holodeck, appeared. Showing incredible potential, such tools are propelling forward how users collaborate, allowing them to be totally immersed in a virtual room, moving within it and having the freedom to closely inspect data not easily represented in the real world; for example, a three-dimensional representation of a geographical region.

One of the potential uses for VR is in crisis management. In Portugal, the most common crisis situations are the wildfires. In a country with a great area of vegetation mixed with the dry and hot summer climate, fires are bound to spark. When these situations happen, multiple teams work to mitigate and ultimately solve them. The focus is on the experts in the command center, trying to understand how the situation is progressing and how to best approach its resolution. These kinds of operations tend to involve high amounts of data and it can sometimes be hard to coordinate the resources on the field with the available access points which can vary over the course of the operation.

This thesis develops a virtual reality system where professionals can share the same virtual room while analyzing, discussing and arriving at approaches to mitigate a crisis scenario, particularly a wildfire. The experts have at their disposal a set of tools allowing them to, among other things, visualize the affected area, spatially and temporally annotate the terrain, command vehicles, get insights from data viewed using the 2D and 3D maps.

1.2 Research Questions

Crisis management is an inherently collaborative endeavour while also requiring visualization of the affected region and relevant geographical details. These facts contributed to it being adopted as the underlying subject explored in this thesis. Focusing on those facets of crisis management, the following research questions were formulated to guide development of the proposed system.

The main broader question is:

- Q - Can a crisis management scenario be improved in a collaborative VR environment?

The main goal is to have multiple users collaborating in the same VE, simulating a shared space where users can work together to develop a strategy and overcome a challenge. This question intends to explore how users can share knowledge within the VE and convey information to each other through intuitive VR interactions, using a notification system that guarantees one user will not miss information highlighted by another. Georeferenced information also shares the spotlight with collaboration in a crisis management scenario. As such, this thesis also explores these kinds of visualizations, using a dual-map interface showing both a 2D and a 3D perspective of the region, and having several tools that interact and extract information from them.

The secondary questions are:

- Q1 - Are there benefits to a dual-map (2D/3D) visualization?
- Q2 - Is there a preference for one interface over the other?
- Q3 - How can users notify each other of new information in the system?
- Q4 - Are users with no VR experience able to adapt quickly to the VE and the available tools?

These secondary questions motivated the design of the system. Question *Q1* and *Q2* informed the implementation of the dual-map interface for map visualization, as well, the design of the VR tools. Question *Q3* motivated the implementation of a *Notification System*, so users can inform other users from within the system about newly available information. It is necessary to keep in mind that not everyone has experience with VR, it is then important to understand how quickly new users can adapt to a collaborative VE. Finally, question *Q4* guided the implementation of intuitive tools to interact and extract information from the dual-map interface, these tools are; *Marker Tool* allowing spatiotemporal tagging, *Brush Tool* allowing drawing on the maps, *Vehicle Tool* allowing commanding of vehicles, *Ruler Tool* allowing measuring distances and the *Wind Tool* allowing control over the wind visualization.

Regardless of use case, there are fundamental actions that are necessary in a collaborative VE, such as the ability to point a collaborator to a specific location or the ability to quickly identify where the collaborator is looking at.

Alerting other collaborators to new information can be done through visual means and, if the VR equipment supports it, through haptic feedback on the controllers. Combined, both approaches can prevent new information from being missed.

1.3 Objectives

The main objective of this dissertation is to:

- **Develop a VR prototype to explore how users can collaborate in a VE.**

For that exploration to be meaningful, the context of a crisis management scenario was adopted. Following is an outline of the steps taken to design and develop the system that will help answer the posed Research Questions:

1. Familiarization with VR techniques for movement, interaction and best practices, as well as the state of the art regarding collaboration in VR settings;

   This first step helps understand what are common actions done in VR and what are users more accustomed to, for example: locomotion through teleportation or haptic and visual feedback when interacting with objects that are interactable. Since collaboration and communication in VR take many forms, researching them helps understand and design features that should be included.

2. Develop a VR prototype. Development of the prototype itself took most of the time of this dissertation. Its development was broken down into several steps.

   a) Design and implement a simple VR experience with minimum required features: movement, multi-user and user representation;

      Using Unity and their XR Toolkit an initial scene will be developed, then the Mirror network framework will be integrated allowing multi-user support.

   b) Architect the system behavior to support synchronization and interaction between multiple users and their actions;

      The shared scene allows for interactions coming from different clients, which must be replicated in all remaining clients. A simple architecture must be designed to allow for quick iteration on the tools.

   c) Process geographical data to extend the system, adding the two maps and layers that can be enable/disable by users;

      Different kinds of geographical data (heightmaps, polygons, polylines, etc) must be processed, converted to use the same Coordinate Reference System (CRS) and imported into the application.

   d) Design and implement tools to interact with the VE with focus on single user experience;

      The first step on tool development will focus first on single user environment and how it interacts with the world and data available.

   e) Adapt previously implemented tools to support collaborative features;

      Then, the second step will adapt the tool to be used in a multi-user environment, making sure its impact is communicated to the remaining users. These two steps will be repeated for as many tools as implemented.

3. Elaborate user tests and evaluate the final state of the system, discussing the results;

   User testing the system is fundamental to understanding if the contributions were attained and what are the areas of focus for future work.

These objectives guided the system's development process. A description of the implemented system will be detailed in the following section, and an overview of what the system looks like is presented in Figure 1.1.

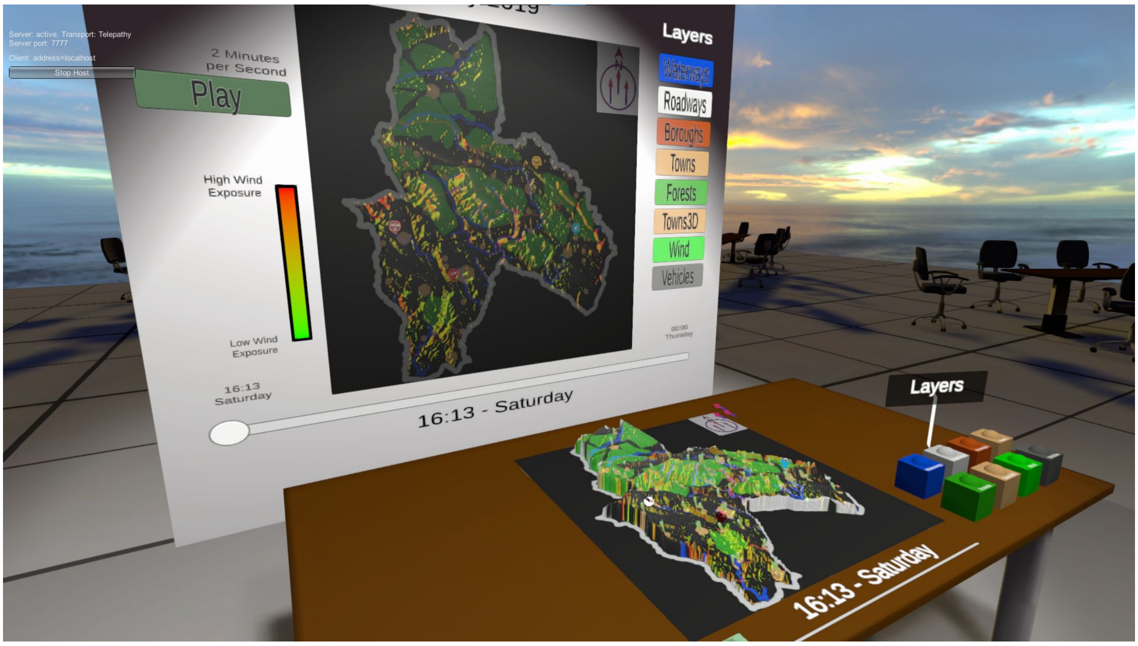


Figure 1.1: Main view and interface. A 2D map on the wall and a 3D map on the table with multiple enabled layers.

1.4 Solution Overview

A system was developed, following the previously stated objectives, in order to answer the previously stated Research Questions. This system implementation will be detailed over the course of this document, but a small overview is presented here.

The VR application consists of a small room with two maps representing a geographic location where a wildfire took place. The maps differ in dimensionality, one is a traditional two-dimensional map, the second is a three-dimensional representation of the terrain. The users can connect to this room through the network, provided one user acts as the host. Once within it, they have at their disposal several tools, seen in Figure 1.2, to annotate, draw and take measurements on the maps, select and command vehicles, and visualize pertinent geographic information such as roads, water courses or towns. These tools are self-contained, each one relates to one aspect of the application and there are no interoperability between them. For example, once the user selects the brush tool, they can only interact with the painting canvas and do not interfere with the remaining tool systems. Users can also control the playback of a real wildfire that occurred in the region - being able to visualize active fire fronts and their respective category and also the total burnt area. Some actions, such as annotating the map, can notify other users and be accepted or rejected based on their feedback. There is also light customization options for some tools, the drawing tool being the prime example.

The system was developed with modularity in mind allowing for testing multiple functionalities and contexts related to maps and geographical data, multi-user VR interactions and crisis management scenarios, using either a VR headset or a normal keyboard

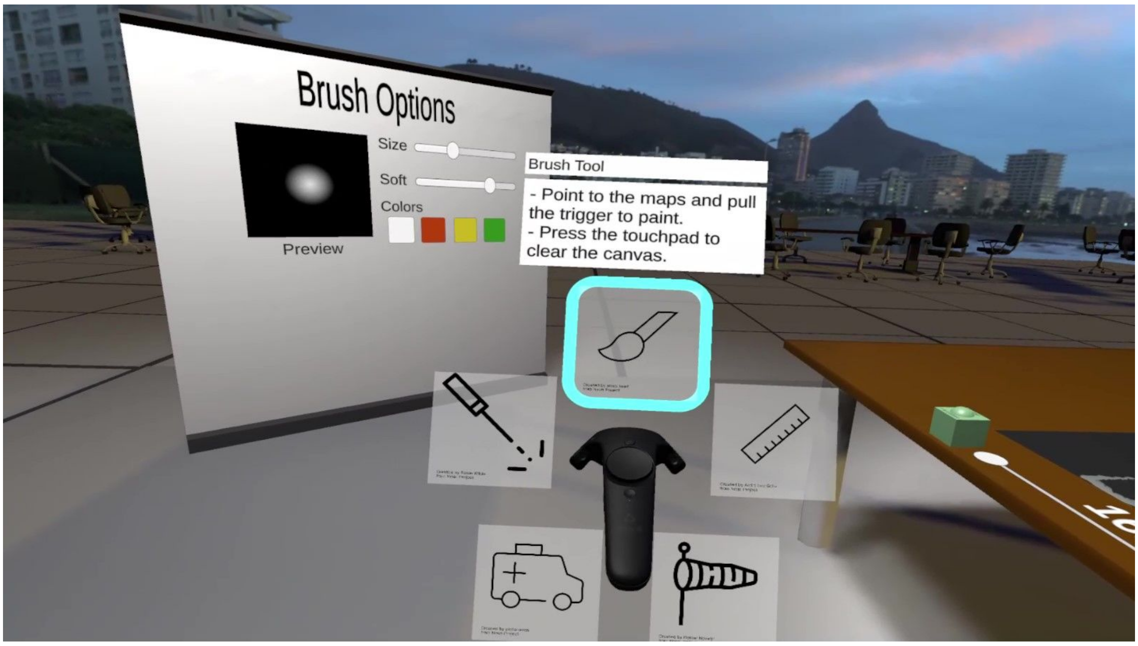


Figure 1.2: Tool wheel attached to left-controller. Brush tool and description highlighted along with brush customization in the background.

and mouse. New tools and features can be quickly implemented and adapted to the network environment; a different scenario with geographical data can also be quickly adapted, after a small amount of data preprocessing.

With these features implemented it is possible to create user tests to help formulate answers to the stated Research Questions.

1.5 Contributions

This thesis contributes to the field several relevant elements. These are summarized below:

- The design and development of a modular VR prototype (which included a thorough review of the current State of the Art, as well as extensive documentation of said prototype) allowing for exploration of various themes in map interaction, data visualization, VR collaboration and crisis management;
- A novel approach in visualizing map data with a dual-map interface presenting geo-referenced data;
- A user study elaborated to test said system with 18 participants where 50% had previous experience with VR and the remaining 50% did not;
- An online repository of self-contained example projects[3] explaining how some of the features were implemented and how some data was obtained/generated.

[3] VRCollabCrisis - Gitlab Example Projects - Last accessed: 13/09/2021

Publication Submissions A poster was submitted to and accepted by the ACM VRST[4] conference and a scientific paper was submitted to the IEEE VR[5] conference and is waiting for review.

1.6 Document Structure

From here on, this document will detail the development process.

Starting with a review of the state of the art of virtual reality research and collaboration in the Chapter 2. Chapter 3 will present a detailed overview of the system; requirements, geographical data used, interaction and collaboration tools, and architecture. Following this description, Chapter 4 will present details on the implementation, stating which technologies were used, then what data was available and how that data was processed to fit the system's needs, followed by implementation details on the interaction and collaboration tools. How the system is structured to guarantee synchronization between the users is also detailed here. Chapter 5 can then present the evaluation through system and user testing. The last chapter, Chapter 6, concludes this dissertation, proposing interesting avenues that can be pursued further.

[4] VRST Conference ACM Symposium on Virtual Reality Software and Technology - Last accessed 15/08/2021

[5] IEEEVR Conference IEEE Conference on Virtual Reality - Last accessed 15/08/2021

Chapter 2

Related Work

Virtual environments have immense potential to improve on conventional collaboration methods. The ability to share a space with another user that might be in another part of the world can remove the need for travel of team members. Real-time visualization of product models at real scale can catch errors in the design at a stage where they aren't as costly. This chapter will survey the history of VR and research into collaboration and interaction techniques within a virtual environment. After that, fields impacted by this tech will be explored and terminating with health issues that arise from being immersed in VR and design guidelines that help prevent these issues.

2.1 History and Technology

In this section, a brief look at the evolution of VR will be taken, since the first prototype in the late 60s, through some of the successes and failures in the 90s, the "VR Winter" in the 00s and the current VR generation started by Palmer Luckey with the Oculus Rift announcement in 2012.

2.1.1 VR History

Virtual reality has been an explored filed, in some way or another, since the late 60s when the first head-mounted three-dimensional display was developed by Ivan Sutherland [41]. Although early, this first implementation already supported positional and orientational head tracking within a 1.8m diameter and 0.9m height volume. The team experimented with two different techniques to achieve this, one based on a mechanical sensor and other, more akin to what is available today, based on ultrasonic wave transmitters and receivers.

From then on up until then 90s, VR applications were mainly developed for military training purposes, particularly flight simulators, as it would bring training costs down.

Then, in the 90s the first commercial VR headsets were starting to appear. Sega launches its first VR headset, Sega VR-1[1], halfway through the decade, which was a part of a rollercoaster-like experience in an arcade setting. The infamous Virtual Boy was release one year later, in 1995. Nintendo's foray into VR was not as successful as they were hoping and was considered a commercial failure. Zachara et al. [44] made a case study out of it, outlining its shortcomings.

The next decade, known as the "VR Winter", did not witness much VR development and interest. It wasn't until the early 10s that the current VR boom was launched with the first prototype of the Oculus Rift designed by Palmer Luckey. This development gained attention and soon after, VR had the attention of big companies, Facebook bought Oculus, Valve partnered with HTC to develop HTC Vive, the first VR system with tracking technology aided by wall-mounted sensors and their Lighthouse technology. Concurrently, Sony was also working on its VR solution, Playstation VR. Google joined in by announcing the Google Cardboard, a low-cost solution for mobile-based VR. By now, almost every big company has or is working on similar technology.

The second chapter in the VR Book [20] has a comprehensive and in-depth review of the history of VR.

2.1.2 Technology

Currently, there are a several companies producing and distributing consumer grade HMD. This comparison will focus on current-gen hardware.

Oculus is commercializing three different HMD, seen in Figure 2.1. They all use inside-out tracking, this means the user does not need external sensors for tracking to occur and consequentially the play space is not restricted to an area covered by sensors.

On the lower end, there is the Oculus Go, Figure 2.1a which is characterized as "All-in-One VR Viewing" meaning no need to have an external device to link the HMD to, such as a computer or, in the Google Cardboard case, a capable smartphone. Oculus Go has one controller and it can track the user's head orientation only, not position, in other words, it has 3DoF tracking. For mid-range, the Oculus Quest, Figure 2.1b, is another

[1] VR-1 Sega's Virtual reality rollercoaster. - Last accessed: 28/07/2020

(a) Oculus Go (b) Oculus Quest (c) Oculus Rift S

Figure 2.1: The Oculus line of Head-Mounted Displays.

all-in-one HMD but this model comes with the ability to connect to a computer, thus harnessing the processing power of the machine enabling it to run applications heavier than those supported by the previously mentioned Oculus Go. The headset, as well as the two controllers the Quest has, support 6DoF tracking, that tracks not only the orientation but also the position of the user's head and hands. The Quest can also support controller-free operation using hand tracking technology. Launched in late 2019, critics considered it to be a revolutionary headset, earning CNET's Editors Choice and Innovation Award[2]. On the high end of the spectrum, the Oculus Rift S, Figure 2.1c, is available, unlike the other two HMD this one requires a connection to a PC to be utilized. It is able to provide much higher quality experiences compared to the Oculus Quest and also supports two controllers and 6DoF tracking.

HTC currently has two HMD series, the Cosmos and the Pro, unlike the Oculus Go/Quest all HTC models need to connect to a computer, there is no standalone model present. HTC also worked along with Valve on the original HTC Vive and HTC Vive Pro, seen in Figure 2.2a. Both HTC Vive Pro and the top-of-the-line Valve index are present in Figure 2.2. The Cosmos series has two different products, where the main difference is how the system tracks the user. Both have 6DoF tracking, the base Cosmos version uses its six cameras to accomplish inside-out tracking, like the Oculus models. The Cosmos Elite attains outside-in tracking using sensors that enable Room-Scale experiences. These sensors comprise the Lighthouse system which is surprisingly open. Oliver Kreylos, developer of the Vrui VR toolkit[3], has a comprehensive analysis of Lighthouse[4]. The Vive Pro is the only model in the Pro series and is the one that we will be using during development time. It has 6DoF outside-in tracking using the Lighthouse system. All the HTC HMDs support two controllers.

Lastly, there is the Valve Index HMD, seen in Figure 2.2b, completely developed in-house. It has 6DoF outside-in tracking using the Lighthouse system. The big innovation of Index is the controllers it comes with. They support finger tracking technology that

[2]CNET: Oculus Quest reivew Review of Oculus Quest headset. - Last accessed: 28/07/2020

[3]Vrui VR Virtual and augmented reality toolkit. - Last accessed: 28/07/2020

[4]Lighthouse tracking examined Blog post detailing inner workings of the Lighthouse system. - Last accessed: 28/07/2020

(a) HTC Vive Pro (b) Valve Index

Figure 2.2: HTC Vive Pro and Valve Index Head-Mounted Displays.

Table 2.1: Comparison of Head-Mounted Displays.

Manufacturer	Model	Tracking	Freedom of Movement	Room-Scale
Oculus	Go	Inside-Out	3DoF	No
Oculus	Quest	Inside-Out	6DoF	Yes
Oculus	Rift S	Inside-Out	6DoF	Yes
HTC	Cosmos	Inside-Out	6DoF	Yes
HTC	Cosmos Elite	Outside-In	6DoF	Yes
HTC	Vive Pro	Outside-In	6DoF	Yes
Valve	Index	Outside-In	6DoF	Yes

allow for single finger movements and interactions.

The Table 2.1 presents a summary of the previously mentioned head-mounted displays.

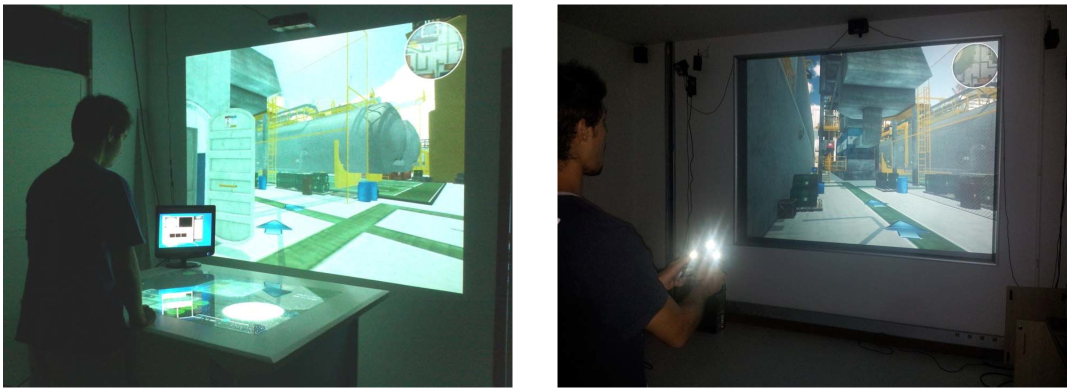
2.2 Collaboration in Virtual Environments

In a virtual scenario, collaboration poses a challenge related to how inter-personal communication manifests. When entering a virtual scenario, where each user's presence is viewed as an avatar, many non-verbal communication clues are lost. A system reliant on collaboration must be able to represent some of these non-verbal clues, such as the users body and head orientations so users can, for instance, understand that they are facing each other.

Medeiros et al. [29] explored collaboration in a 3D virtual environment (not in a head-mounted display context) by analyzing the applicability of the 3C (Communication, Coordination, Cooperation) model of collaboration. They implemented a system supporting two different roles, a *Technician* and a *Worker*, in the context of an industrial use-case. Users views are illustrated in Figures 2.3. The *Technician*, Figure 2.3a has at their disposal a dual view of the scene composed of a common first person view projected to the wall in front of the user, an overhead view of the location akin to a map on the multi-touch table at arms length. This dual view was an initial inspiration for the implementation of a dual-map setup for this thesis. Among other features, the *Technician* can direct the *Worker* to places on the map by placing waypoints in the environment. The *Worker*, Figure 2.3b, is then able to move and interact with the environment using a conventional console controller.

2.2.1 Co-located Collaboration

In Dollhouse VR, Ibayashi et al. [18] came up with interesting solutions to communication between users immersed in VR and users outside. Both classes of users share a space, where one is within, in a first person perspective using the HMD, the other is viewing the world from a top-down perspective.



(a) Technician's interface. (b) Worker interface.

Figure 2.3: Different views from both roles available in Medeiros et al. [29].

The user outside can interact with the inside through a touchscreen interface and these interactions manifest within the world for the user inside to see, as pictured in Figure 2.4. The user in VR can look up through a glass ceiling and see the other user through a live-feed thus being able to capture gaze and facial expressions that ease communication. One thing the user in VR can do to signal back to the user outside is pressing a touchscreen smartphone glued to the front of the headset to point in a given direction, this solution was necessary because the VR experience was seated and using a conventional gamepad controller, not one with 6DoF or even 3DoF.

Vasco Pereira et al. [32] developed a framework specifically geared to support development of collaborative Extended Reality (XR) applications and implemented both Virtual Reality (VR) and Augmented Reality (AR) clients to test said framework. They implemented several effects to help guide users to an highlighted user or object. These were; *Arrow*, *Radar* and *Transparent Walls*, which in one way or another inspired this thesis notification system.

Lages et al. [24] worked on adapting a popular mobile game, Krinkle Krusher, to virtual reality focusing on a co-located experience with natural interactions. The original

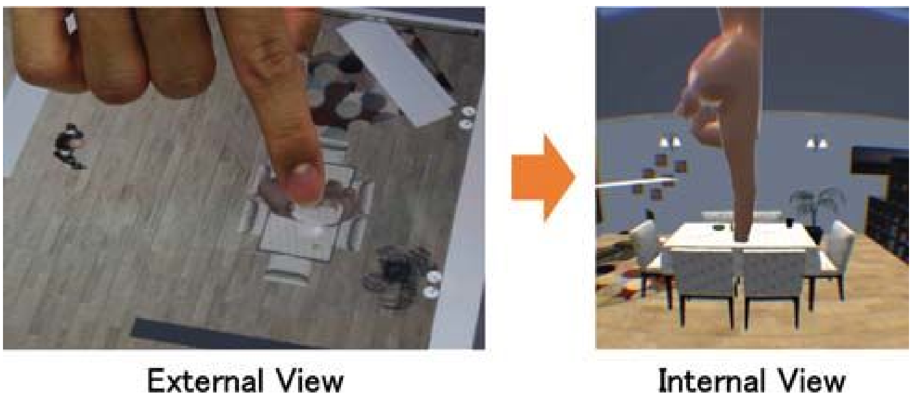


Figure 2.4: The view from outside (left) and inside (right) of the virtual environment [18].

version consisted of a fixed third-person perspective onto the gameplay area, this was then adapted to a first-person perspective in the virtual reality version. The authors decided to confine the play area to a circle within the game world to avoid unnatural locomotion techniques, such as controllers or redirected walking. With this choice, the locomotion technique ended up being natural walking, with the system using OptiTrack [5] hardware to track both HMDs simultaneously. The videogame supported the casting of spells that should be target at enemies. Again, the authors went for natural gestures and by using hand tracking provided by the Leap Motion controller they were able to make the casting of the spells as easy as extending the open palm forwards within the user's field-of-view. The authors also experimented with a shoving gesture to push away close-by enemies but this approach revealed some health concerns as the users would accidentally shove each other. On a social perspective, the users saw each other represented by an avatar with floating, disembodied hands and an simplified face, as seen in Figure 2.5. This solution raises an important problem in VR collaboration, the lack of facial expressions expressed by the avatars as it can be a difficult problem to mirror the users facial expressions on the avatars while avoiding the Uncanny Valley.

2.2.2 Networked Collaboration

Knispel et al. [23] at the University of Western Australia explored non-verbal collaborative painting. Users would click a hyperlink and get taken into a VR world with other users where the only communication afforded was painting and gesturing with the cylinders used to paint, that represent the hands. The researchers observed users organically start

[5] OptiTrack Commercial solutions for motion capture. - Last accessed: 28/07/2020

Figure 2.5: Player avatar with disembodied hands and simplified face [24].

to work together with barely any communication. Their drawings usually converged to a scene from the real world.

The authors noted the anonymity of the experience helped with its enjoyability. They also noted that there were some users mentioning the need for voice communication.

Elvezio et al. [13] explored low-latency interaction between networked users. Their system allowed two users to control a board with four ropes attached to edges. Each user controlled two of the four ropes and by pushing and pulling on the rope that effect would be reflected on the rigid-body board. This kind of interaction would be useful for physical therapy where both the patient and the therapist would have to, for example, move their arms high overhead. From an implementation aspect, the authors used Unity in conjunction with the MercuryMessaging [14] toolkit to develop their solution.

For complex tasks that may need expert input, solutions that use both virtual reality and augmented reality have been explored. Elvezio et al. [12] explored this subject with two users, the expert in a virtual reality setting off-premises and the local user with a see-through HMD. Their particular solutions revolves around the creation of virtual replicas so that the expert can guide the local user in a complex assembly task, pictured in Figure 2.6. As the authors noted, language to point and manipulate objects in 3D space is often ambiguous, thus their virtual replica solution supports manipulation, pointing and annotations in 3D space.

Ardal et al. [1], explored application of virtual reality to filmmaking. The authors developed a tool that allows remote collaborative previsualization, a necessary step in pre-production. The system was designed with three distinct phases **scene preparation**, **realtime animation** and **video export**. The first phase consists of preparing the set where the scene will be visualized. This preparation is not inside the virtual environment and is done using the Unity editor, the tool the authors chose to develop in. The second phase is the main one, where both users inside the virtual environment control character movement and dialog, along with camera motion and optics. The last phase consists of exporting the frames recorded by the virtual camera. Interface-wise, the authors opted for floating in-world menus tied to the position of one of the controllers. One aspect of

Figure 2.6: Local user with HMD (left). View from local user with virtual replica on display (right) [12].

the interfaces worth mentioning is the adoption of asymmetric user interfaces, where users were given different interfaces based on their role, particularly the director, charged with handling actors, and photographer, charged with handling the camera. The authors conducted a within-group study with 20 filmmakers and the results indicate the tool to be useful for collaborative previsualization work.

Conges et al. [10] developed a Common Operational Pictures (COP) implementation in collaborative virtual-reality environment. The COP representation helps aggregating and visualizing data coming form multiple sources, improving situational awareness and collaborative decision making. Their work centers around a map presented on top of a table, seen in Figure 2.7 and the COP is able to consume and display real-time information coming from external sources.

2.2.3 Social Aspect of Collaboration

When users share a virtual space they must feel like they are sharing it with another human being. Smith et al. [39] explored the impact of having a representation of self and collaborators within the virtual environment. The researchers compared how users collaborated in three different scenarios, **face-to-face** in the real world, **embodied VR** where users had an avatar representing their body and their collaborator's and **no embodiment VR** where users shared the virtual space, could interact with it but had no representation of their own body or their collaborator's. The authors conducted a study with 60 subjects, pairs of strangers were formed to complete two tasks. The first was a negotiation task where users would decide which uses to give to rooms on an apartment. Then, the second task was to agree on the furniture placement in said apartment. The authors registered non-verbal (gestures, gaze) and verbal communication between the participants. The study revealed similar verbal and non-verbal communication in the face-to-face and embodied VR scenarios which was not as present in the no embodiment VR scenario. There was a clear preference among the participants to have an avatar represented in the virtual

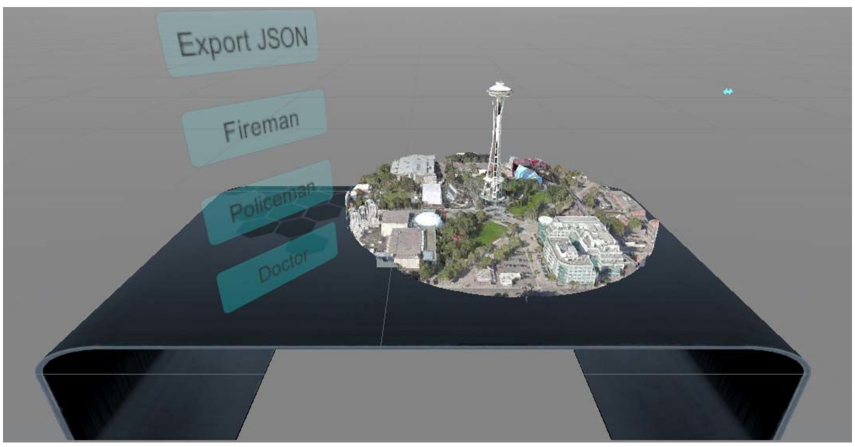


Figure 2.7: Conges et al. [10] map visualization on a table.

environment, they also noted the feeling of being alone in the no embodiment scenario.

Avatars are a common concept in virtual worlds, being it VR or traditional 3D environments, but when used in VR several questions arise. Should the avatar be a full-body representation? How faithful should this representation be, is full-body tracking a solution or are predefined animations a good enough to avoid extra hardware necessary for the full-body tracking solution? Or are disembodied hands and heads an acceptable approximation? Heidicker et al. [17] explored these questions inspired by the different implementations of avatars in commercial products, such as Facebook's social VR, AltspaceVR[6] and High Fidelity[7]. The authors developed a social VR system to compare three different approaches to avatar representation. Avatars with **complete body and predefined animations**, **complete body and body tracking** using the Microsoft Kinect v2 and **only head and hands** represented were tested in a user study. These are illustrated in Figure 2.8. The authors concluded that the full-body tracking method is the method that gives the highest co-presence feeling on the participants. Somewhat surprisingly, the second best solution was the avatar with only heads and hands tracked which isn't significantly worse than the full-body tracking and is better than the avatar with predefined animations.

[6] AltspaceVR Virtual reality meeting place. - Last accessed: 28/07/2020

[7] High Fidelity An audio-only meeting platform. At the time the Heidicker et al. [17] was published, High Fidelity was, in fact, a project focused on virtual reality meeting spaces, since then the product as shifted to be audio-only. - Last accessed: 28/07/2020

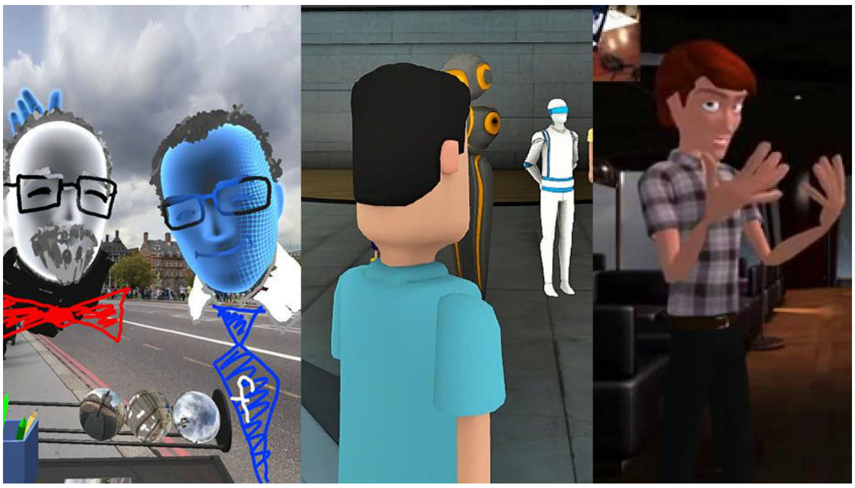


Figure 2.8: Different avatar approaches [17]. Head and hand tracking in Facebook's social VR (left), complete body with predefined animations in AltspaceVR (middle) and complete body with body tracking in High Fidelity (right).

2.3 World Immersive Interaction

When inside a virtual space there are plenty of interactions to be considered. Particularly in a VR setting, where the level of immersion is higher than in a video-game, user expectations regarding interaction are also higher. Interactions with the world, the objects and the menus must be considered, if overlooked these may lead to adverse health effects or breaks-in-presence.

2.3.1 Movement in Virtual Worlds

Movement, particularly locomotion, inside a VR experience is crucial to get right in order to avoid a specific type of motion sickness caused by senses disagreeing on wether the user is moving or stationary, this specific case is called simulator sickness and is discussed later. In order to avoid these adverse conditions, special care must be taken when choosing and implementing a locomotion scheme.

Bhuvaneswari Sarupuri [35] developed LUTE - Locomotion Usability Test Environment and tested multiple locomotion solution. LUTE is a test environment for short, medium and long distance travel in VR worlds, with customizable path complexity. In it, the author tested the more general teleportation method of locomotion for long distance travel, a joystick based locomotion for medium distances and for short distances Trigger Walking was tested. The author also contributed to the development of this last technique [36], which is based on mimicking walking but instead of moving the legs, the user presses the triggers on the VR controllers to move.

The author concluded that one single technique is not suitable for the three different locomotion scenarios. The research compared Trigger Walking to other techniques and found it favorable for short distances as it induced less simulation sickness, less fatigue (although finger fatigue could become a problem) and higher usability score compared to the other methods.

Another consideration to have regarding locomotion is the user's surroundings. Particularly in inside-out tracking scenarios, where the user is not confined to a player area they might stumble into objects present in the room. While developing a novel locomotion technique based on combining Natural Walking and Walking in Place, Sousa et al. [40] observed this combination to be conducive to stumbling. The authors then implemented an augmented reality component to their original solution, converging on CWIP-AVR - Combined Walking in Place Augmented Virtual Reality. This technique allows users to seamlessly transition from walking in place to natural walking and back while being warned by a visual indicator, warning arrows and sound alerts from their AVR system. This system materializes two groups of indicators; translucent planes to bound the limits of the physical room and obstacles, rendered as solid parallelepipeds that happen to be in the user's path. These indicators both change color according to a function of distance to the user. Their user study reported easy adoption of the technique.

2.3.2 Object Interaction and Manipulation

In order to make a VR experience interesting there must be objects that are interactable in some way. The simplest interaction might be grabbing an object and throwing it around, there is also the possibility of the object being manipulated in some way such as changes to its properties. The previous example changes the position and rotation of the grabbed object, but there are other properties that can be changed, general ones would include scale, color, textures along with the position and orientation already mention. Depending on the specific application other kinds of properties could be manipulated. These kinds of interactions/manipulations are common-place in VR experiences and there are already some patterns for them, defined in chapter 28 of the VR Book [20]. Focus will be on the **Selection** and **Manipulation** patterns.

The most common **Selection** patterns are the **Hand Selection Pattern** and the **Pointing Pattern**, detailed below.

The **Hand Selection Pattern**, as the name hints at, uses representations of the user's hands in the VR space, its the most intuitive way to pick up an object as it mimics exactly what people do in the real world, extending the hand and then grab an objects, this grab gesture could be a trigger press, for example. A question arises when representing hands, should they be as realistic as possible or abstracted? There are some approaches that represent hand and arm (through inverse kinematics), disembodied hands and abstracted cursor that track hand placements, an example of these representations can be seen in Figure 2.9. Any of these solutions are acceptable from a user perspective and don't necessarily break the illusion, although care must be taken when using inverse kinematics to represent the arms as they might not represent the real positions. This technique is a great solution for objects close to the user and becomes unfeasible for distant objects. An extension to this method would be the **Go-go Technique**, that behaves normally in most circumstances but changes when the users extends their arms to full length, gradually increasing the reach of the virtual hand to reach objects further away.

The **Pointing Pattern** is frequently used in mobile based VR, where hand representation is more limited. It is based on extending a ray either from the user's point-of-view or the user's hand, then, the first object the ray intersects is the object that will be selected upon a trigger pull or in the case of mobile VR, using something like dwell selection.

Figure 2.9: Realistic hands (left), hands with no arms (center), abstract hands (right) [20].

Dwell selection requires the user to leave the ray pointing at an object for a small amount of time for that object to be selected. This solution can make it hard to precisely select objects far away due to small hand movements transforming in large movements at the tip of the ray. There are some techniques that can be applied to reduce this problem such as snapping to the most important object closest to the ray, with importance being previously assigned, or using two hands to aid pointing, where one hand controls large magnitude movements and other controls smaller, finer movements.

The patterns presented are not that suitable for selecting multiple elements. To address that situation there are **Volume-Based solutions**. Modifying the **Pointing Pattern** to casting a cone instead of a ray and be able to control the radius would be a simple solution to selecting multiple objects that are close to each other. Other more complex solution would be a box-based solution, where the user manipulates a 3D box to enclose all the objects to be selected.

Regarding **Manipulation**, the **Direct Hand Manipulation Pattern** is very similar to the **Hand Selection Pattern** in the sense that it is very similar to what is done in the real world. After grabbing the object, the users rotate and translate the object with their own hands as if they were doing it with a real object. The **Proxy Pattern**, uses a proxy object that represents the object to be manipulated, this proxy object can be physical or can be a smaller representation of an object bigger or further away from the user. The user would manipulate this proxy and any transformations to it would be applied to the proxied object.

These patterns can be combined into a new one, a particularly interesting combination is the **World-in-Miniature Pattern**. This pattern gives the user a view over the world they reside, they can see the space they are in, themselves and the objects that share that space with them. They could also select and manipulate objects from that miniature representation, they could also grab and reposition themselves within that map in a teleportation like transition.

2.3.3 In-World Graphical User Interfaces

Graphical user interfaces, commonly known as GUIs, are commonplace in many 3D applications, usually implemented as a 2D layer on top of the 3D world not directly part of it. There have been some 3D applications, mainly games, that do put the GUI in world, but mainly for artistic effect, not as necessity which is the case for VR experiences. These different ways of approaching UI design are known as **diegetic**, when the interface is entirely part of the world, and **non-diegetic**, when that is not the case. According to most research in UIs for VR, the diegetic way is preferable over the non-diegetic way, as the former can help with immersion, thus contributing to less breaks-in-presence, these two concepts are explained below, in Section 2.5.1.

Azai et al. [2] proposed the Open Palm Menu, a technique to display a horizontal menu anchored to the palm of the user's hand, as seen in Figure 2.10. The menu would

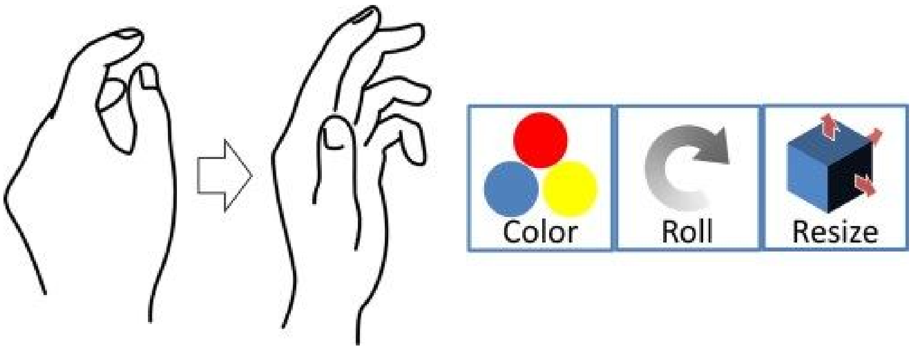


Figure 2.10: Rotating the wrist to show the menu [2].

open when the user oriented the palm to be parallel with the user's line of sight and the menu would extend from the palm. Once the user was done with the menu, they would turn the hand 90 degrees so as to make the palm face away from the user's face and the menu would close. The authors tested two types of menus where the elements would extend in different directions, vertical or horizontal. In their tests, they noted that most users prefer the vertical menu over the horizontal one.

Azai et al. [3] also proposed the Tap-Tap Menu, an implementation of a menu interface attached to the user's body. The authors used hands, forearms, abdomen and upper legs as the display areas. The user's body is an interesting place to put menus as tactile feedback can be obtain without the need for external objects and menu options can also be navigated to through kinesthesia, our way of telling where our own body parts are even with eyes closed.

There have been multiple body-based solutions for menu placement in virtual environments. Lediaeva et al. [25] elaborated a comparative study of body-referenced GUIs in virtual environments. Their research also provided important design guidelines for implementation of such menus, as the technique is an open research topic there aren't many best practices defined. The authors compared menu placements on the hand, arm, waist and in the virtual world, they also tested two menu shapes, linear and radial, and three selection techniques, based on raycasting, head-tracking and eye-tracking. Their results show which combinations of the considered scenarios best performed and which were preferred by the users. Specifically, they found that spatial, hand and waist menus are faster than arm menus. Regarding selection techniques, the authors did not find significant differences besides the eye-tracking method being more prone to errors even then, it did not affect task completion time significantly. Lastly, the participants ranked the spatial menu their favorite placement and the arm menu their least.

2.4 Current Applications of Virtual Reality

From full-length video-games and short experimental movies to commercial products for medical education/training and large scale data visualizations systems, uses for VR technology are not scarce and show no sign of slowing down.

2.4.1 Crisis Management and Response

Virtual reality has been used to prepare and train for crisis scenarios. The EGCERSIS [9], illustrated in Figure 2.11 aims precisely for this, recognizing that training for such scenarios are difficult to recreate for training purposes, thus making the training scenarios not as realistic as they could be. Also, the logistics behind organizing a training scenario make it impossible to do frequently, which in turn lowers the amount of scenarios that can be efficiently practiced. The authors propose a system to address these shortcomings and improve on the data gathering of these procedures. The focus of the system is on four components, according to the article:

- Exercise editor (modeling and scenario definition);
- Virtual environment (multi-player scenario enactment);
- Decision support tools (accompanying decision-maker of the crisis cell);
- Dashboard tools (real-time and later analysis of players, exercises and plans).

For a scenario to be created and used within the system, first a virtual twin of the physical space must be made with the assistance of maps and building plans, the authors modeled a metro station seen in Figure 2.11a. Then, site managers, responder and authorities will use the Exercise Editor to model a dynamic scenario that can change based on the actions and decisions of the responders. At this point, everything is ready for the responders to jump into the virtual environment and will take on the scenario, in Figure 2.11b we can see a first responder (firefighter) extinguish a fire using a tool. Meanwhile, the crisis management team is assessing the situation with the Decision support tools. Throughout this exercise, the movement and decisions of the responders will be recorded so that after the scenario is complete the recordings can be replayed and reviewed which should dramatically improve how the coming exercises play out.

Sermet and Demir [38] presented a virtual reality system for disaster response that supports multiple use cases, illustrated in Figure 2.12. It supports multiple clients connected simultaneously to a web server using different devices, which include VR headsets as well as mobile and desktop clients. A very interesting feature of the system is the data retrieval and dynamic scene generation allowing flexibility to tackle different case studies with less effort, contributing to a more generalized solution. Of note, the authors present the geographical data as if it were on top of a table.

While not being related to virtual reality, the work from Döweling et al. [11] explores collaborative crisis management using interactive tabletop screens, as seen in Figure 2.13. Their system supports multi-touch input as well as pen input to interact with the tabletop. It allows users to import data from publicly available services and make annotations on the data. Another interesting feature of the system, is that it enforces a hierarchy of users based on roles and can present role specific data and restrict access to features based on the user's role.

2.4.2 Entertainment, Culture and Tourism

In entertainment, VR has a big presence in the video-game industry. Looking at Steam, a popular storefront for PC video-games, and searching for VR-Only titles around 4300 results show up all marked as VR-Only, this does not necessarily mean they are all big VR projects, some may be small experiences or even desktop applications. That already gives an idea of the amount of existing titles. And this is just the titles on this one platform, excluding the ones that are exclusive to the Oculus, Vive and Playstation HMDs. The movie industry is also experimenting with using VR and AR in their productions. In 2018, Walt Disney Animation Studios released **Cycles** [15] at SIGGRAPH, it is the first short set in VR. In May 2020, a complete AR experience featuring Wallace and Gromit[8] was unveiled and is set to be released late 2020. Although, there aren't many VR titles to talk about in the movie industry, the technology itself has been gaining traction on the production side. Most recently, the 2019 movie Lion King, a live-action recreation of the homonymous classic was filmed and directed in virtual reality.

In the culture and tourism aspect, there is one experience that must be mentioned. The giant Google [22] launched, in late 2016, Google Earth VR, pictured in Figure 2.14. The system was presented in a SIGGRAPH 2017 talk, where the authors detailed the techniques they developed, for instance, ways of navigation at wildly different scales across the globe.

Other more focused cultural experiences aided by modern photogrammetry techniques have been explored. Relieve History [43] is a project that allows users to experience the Ayutthaya historical park in Thailand, as it is now and how it was originally

[8]Wallace & Gromit: The Big Fix Up - Recruitment Drive Wallace & Gromit Trailer. - Last accessed: 28/07/2020

(a) The metro station modeled.

(b) A first responder using a extinguishing a fire.

Figure 2.11: A look inside the virtual environment modeled for crisis exercises [9].

built. The present model was obtained through photogrammetry and the past was reconstructed based on the model and historical accounts and documents. On top of this, the authors implemented role playing missions that would teach the users about the ancient culture of the World Heritage Site. It is also interesting to take a look at AR and its uses in the tourism sector. Nóbrega et al. [30] investigated how can augmented reality techniques complement the traditional paper map by adding tri-dimensional meshes and georeferenced data ont top of it. The system serves as a base graphical layer that allows different kinds of applications to superimpose their data on top of a physical map.

2.4.3 Education

In education, VR is very prevalent in fields where access to case studies can be difficult, such as the Medical Industry. For this reason, researchers have developed multiple solutions to aid medical training and education.

PathoGenius VR [28] is a virtual classroom where a medicine student can practice within a virtual environment generated based on instructor's input. The system has one interface for data input in which the instructor sets the parameters for the scenario that needs training. The base scenario is an exam room where the patient goes to have a diagnosis, challenges are then dynamically generated based on the instructor's parameters. The second interface is the student's, it provides access to the patient's vitals and a way to administer X-rays and blood tests. The student can make these actions through the use of virtual tools such as stethoscopes, thermometers and syringes, for example. The authors conducted a preliminary study with eight students, when asked to choose between PathoGenius VR and the common written case they would choose the VR experience.

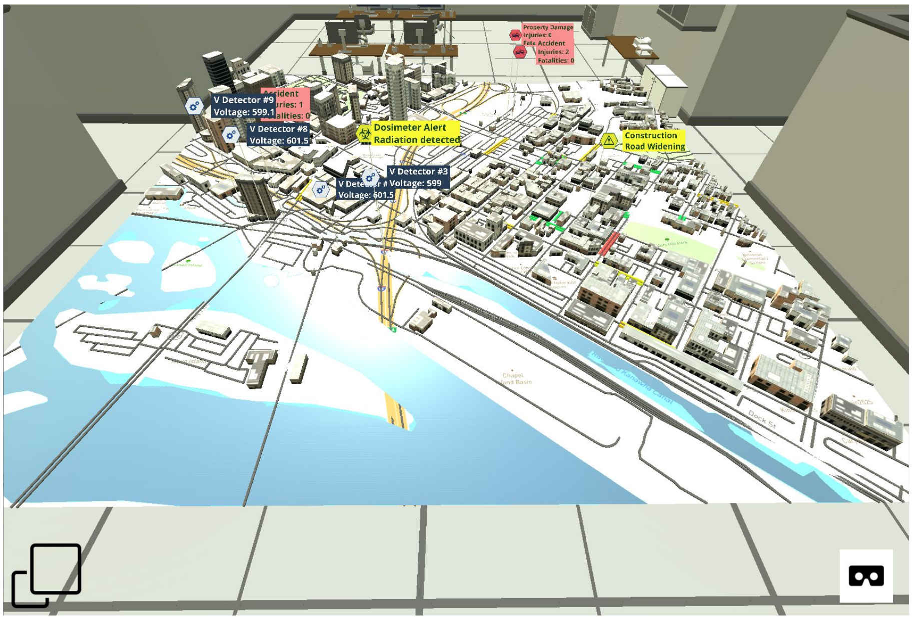


Figure 2.12: Dynamically generated scene with overlaid data from sensors. Sermet and Demir [38].

Figure 2.13: Döweling et al. [11] interactive tabletop in use.

Still in the medical field, training and educating surgeons is also a challenge. Traditionally, soon-to-be surgeons train on cadavers, through book and through a master-apprentice relation with a master surgeon. ORamaVR [31] developed M.A.G.E.S a training and educational solution for surgeons, pictured in Figure 2.15. The team is able to support multiple networked individuals in the same virtual environment through a custom Conformal Geometric Algebra GPU interpolation engine that helps reducing the data that needs to be transferred between the users. The authors also offer an analytics suite, a Software Development Kit to prototype and model new scenarios and it also integrates an educational curriculum that the user can use as reference while training.

2.4.4 Interactive Data Visualization

Other interesting use of VR is in data visualization systems and exploratory data analysis. As the amount of data produced by all the systems, applications, websites rise, so too the need for understanding and interpreting this data. These tasks fall on the Big Data field, but it also raises a complementary problem, that is, how to visualize all this data.

Figure 2.14: Google Earth VR [22].

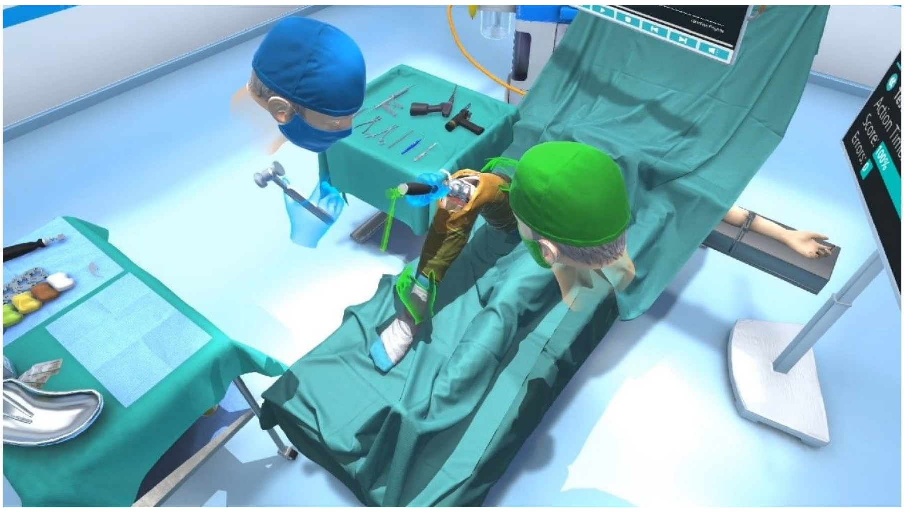


Figure 2.15: M.A.G.E.S system [31].

Commercial products are already available such as syGlass[9].

VRoadworks [42] is a data visualization prototype to be used by traffic engineers and similar experts to coordinate construction work, in the city of Hamburg, which often times affect traffic. The team ended up designing an interactive 3D visualization of multidimensional dependencies. The solution presents multiple layers of maps representing different times (present, near future, far future) and it also presents basic interaction techniques. **Selection of sites** was implemented with an out of the box feature of the Oculus Touch controllers, the user extends their index finger and touches the construction site. Upon selection, the user receives haptic feedback indicating the selection was set. **Zooming and rotating** the map were implemented with the **Direct Hand Manipulation Pattern**, but with two hands instead of just the one. This allows the user to manipulate the map in a more familiar way, the standard way to interface with maps on current smartphones. Moving the map is achieved by grabbing and moving it, in order to rotate it the user must move the hands in a circular motion, while the map is still grabbed and to zoom it the user moves their hands closer or further apart.

The authors conducted a user study with seven participants and a test with a roadworks expert to uncover usability shortcoming in the system and the response was positive by all participants.

Caballo et al. [7] developed Immersive Insights specifically for collaborative exploratory data analysis. This application is running on top of a very intricate system, named Dataspace. Dataspace is a hybrid reality system, it is room-scale and reconfigurable, as seen in Figure 2.16. This system is composed of:

- 15 UHD displays with touch capability, which can be moved and rotated in space via robotic arms mounted to the ceiling;

[9]syGlass Scientific data visualization tool. - Last accessed: 28/07/2020

- A central table onto which visual content can be displayed through two HD projectors; Touch and gestures performed on the table are detected through a set of eight Kinect v2 sensors;
- A spatial audio system consisting of 20+2 speakers, and an array of four directional microphones that can be used to detect voice commands and their sources;
- A set of augmented reality headsets (currently Microsoft Hololens and Magic Leap One devices) to interact with spatial or high dimensional data, often visualized atop the central table;
- A set of virtual reality headsets (Samsung Odyssey) to remotely access the environment and its functionalities, providing a virtual replica of Dataspace and its content.

It can operate in multiple modes within the real-virtual continuum. It can be a fully physical system, an hybrid system using the augmented reality headsets mentioned above and the physical components, completely in augmented reality or completely in virtual reality. Supporting these modes make this system exceptionally useful for direct comparisons between them. Thus, Immersive Insights compare the preferred way to do exploratory data analysis. To understand how the VR and AR approaches to this analysis fare against the physical methods, the team performed an initial two-part user study with twelve data scientists. The first part of the study was focused on how much the VR/AR components would help or hinder the scientists individually. The second part was focused on collaboration between the scientists and comparing it to desktop-based tool. The authors results agree that integrating augmented reality into exploratory data analysis tools can have improvements in task duration. For the first part of the study, they also observed that standalone virtual and augmented reality underperformed compared to the hybrid solution. The second part of the study showed the users in Immersive Insights arrived at more insights in less time than the users working in the desktop-based tool. Indicating that, in fact, augmented reality tools can improve current exploratory data analysis solutions.

Figure 2.16: A Mixed Reality look at Dataspace [7].

The research work of Coffey et al. [8], also contributed inspiration to the dual-map design implemented in this thesis. The researchers developed the Slice WIM, an interface to visualize three-dimensional volumetric data, seen in Figure 2.17. It provides the user with three different views over the subject, a 3D view and two slices of the volumetric data, one projected on the wall in front and the other projected to the table below. The user has control over where the data is sliced in an interactive fashion allowing for interesting exploratory data analysis, using the interactive multi-touch tabletop interface.

2.5 Health Issues and Design Guidelines

Part of what makes VR experiences so engaging is how it can make the user feel part of a virtual world, being it a realistic copy of our own or inaccessible, impossible ones. Two concepts contribute to the understanding of this feeling of belonging, these are **presence** and **immersion**.

Due to the feeling of actually being there, VR experiences are susceptible to causing health problems. From general discomfort using an HMD to the possibility of a user tripping and falling, designing a VR experience requires some health related awareness. To aid development, there are some useful references that can be used as VR guidelines.

2.5.1 Presence and Immersion

Immersion and presence are concepts somewhat intertwined, one definitely influences the other.

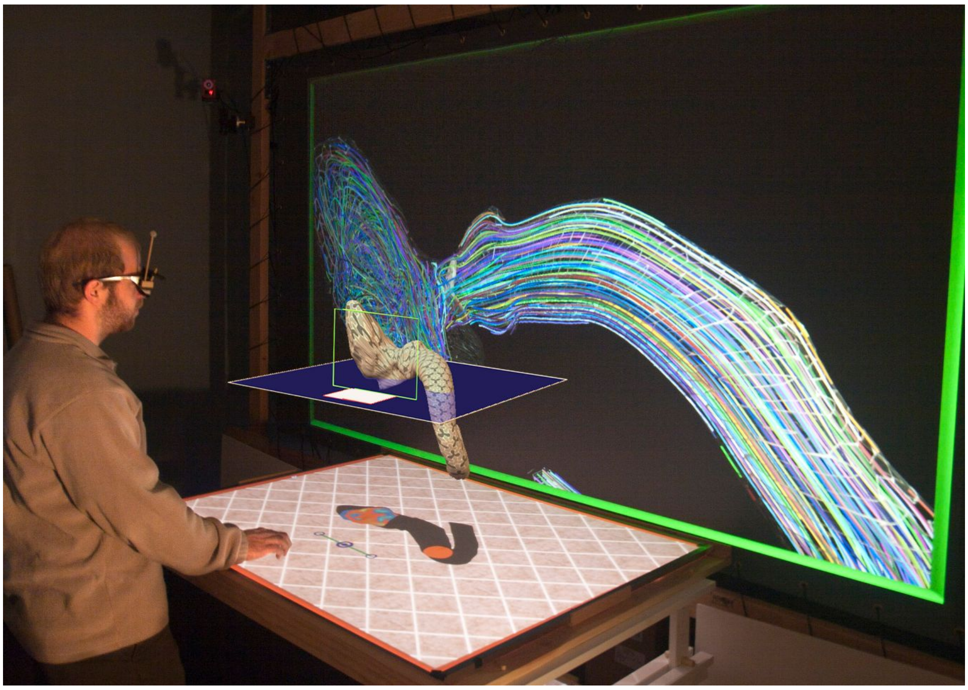


Figure 2.17: Interactive exploration of volumetric data using Slice WIM. Coffey et al. [8]

According to chapter 4 in the VR Book [20], immersion is the objective degree to which a VR system and application projects stimuli onto the sensory receptors of the user in a way that is **extensive** - in the range of stimulated senses - **matching** - the head motions, body tracking and other sensory modalities - **surrounding** - panoramic sensory clues - **vivid** - high-quality resolution, lighting, framerate - **interactive** - ability to interact and affect the world - and **plot informing** - in a way that complements the story the experience is telling.

The International Society for Presence Research defines presence as "... a psychological state or subjective perception in which even though part or all of an individual's current experience is generated by and/or filtered through human-made technology, part or all of the individual's perception fails to accurately acknowledge the role of the technology in the experience." [19]. This state, this feeling, where the user fails to perceive the technology, is what VR is able to excel at over all other kinds of technologies.

Thus, presence is an illusion facilitated by immersion. And, as all illusions, it can break. These breaks in presence are undesirable in any virtual reality setting as they can completely take the user out of the experience. Some illusions of presence are documented along the chapter 4 of the previously mentioned book, each of them important to understand in order to create experiences that take advantage of and avoid breaking those necessary illusions. These illusions are:

- Illusion of Being in a Stable Spatial Place;
- Illusion of Self-Embodiment;
- Illusion of Physical Interaction;
- Illusion of Social Communication.

These illusions related to how the users see the world, see themselves, feel that the world responds to their actions and feel that they exist in the social space, i.e. the other social entities, A.I. agents or other users, communicate and acknowledge the existence of the user.

2.5.2 Simulator Sickness

Simulator sickness (SS) is one of the most pervasive and more widely studied affliction caused by VR. There are multiple factors that can cause this motion sickness related problem and multiple ways to minimize it, but, being a physiological problem that differ from user to user, there is not one given solution for it, only strategies to minimize it as much as possible.

According to David Johnson et al. [21] SS has been a known phenomenon since the mid 50s, when the U.S. army first started helicopter flight simulations.

SS can cause symptoms such as headaches, nausea and disorientation. According to Gerard Llorach et al. [26] the most widely accepted cause of SS lies in sensory conflict theory. Situations where what the user senses in the virtual environment is not sensed in the real world might provoke SS, one such situation would be the users seeing themselves moving in VR but staying stationary in the real world. Their results corroborates this fact as the users that moved in VR with a common game controller reported higher SS then those using the position estimation system developed by the authors [33].

The major contributor for this disparity is **Latency**. Other contributors are:

- Duration of the experience;
- Field-of-View;
- Interpupillary Distance;
- Position-Tracking Error.

Latency is an undesired effect in many contexts, but particularly in VR as it can cause breaks in presence and, as stated previously, it can have a great impact on wether someone experiences simulator sickness or not. The VR Book [20], that dedicates the whole chapter 15 to this subject, defines latency as "the time a system takes to respond to a user's action", meaning the time between the user making a gesture and the time that gesture is represented on the system. Higher latency was a major problem in early HMDs and improvements in this front, mainly by Palmer Luckey at Oculus, is a part of what brought the current VR boom in recent years. Sources of latency are:

- Tracking Delay;
- Application Delay;
- Rendering Delay;
- Display Delay;
- Synchronization Delay.

An in-depth examination of these sources can be found in the previously mentioned chapter of the VR Book.

2.5.3 Design Guidelines

When developing any kind of software meant to be interacted with, it is useful to have some design guidelines to reference in order to avoid developing inferior solutions to problems already solved.

These design guidelines are usually product of extensive user research and developed over long periods of time. One such example would be Apple's Human Interface Guidelines[10] that help developers achieve a higher quality and better user experience for their

[10] Human Interface Guidelines Apple's design guidelines for their platforms. - Last accessed: 28/07/2020

products and, if programmers fail to get AppStore certification, they are usually pointed to the guidelines to help them with their next submission.

For VR, similar resources have been developed but haven't had enough time to mature yet. As of this writing there is one main resource for VR design guidelines, that is TheVRBook [20]. According to its website, it provides more than 600 applicable guidelines for VR developers and is definitely a good reference to have for anyone working in the field.

The author presents high-level guidelines at the end of each of its six parts summarizing the in-depth explanations previously presented.

Another good reference, more akin to Apple's than book's, is Oculus' VR Design Best Practices[11]. Maintained by one of the main contributors to the second wave of VR, these guidelines are much more direct and implementation focused compared to the ones presented in the book.

Other helpful resource is the Designing for Google Cardboard[12] along with its companion app Cardboard Design App[13].

2.6 Summary

The chapter started with a small review of the historical context surrounding virtual reality, starting in the sixties with the first prototype of a VR headset and ending with a survey of consumer head-mounted displays available in 2020. Then, insights from other researchers into collaboration techniques were presented with focus on non-verbal communication. Following, interactions techniques and patterns were detailed along with research into GUIs. Current applications of VR technology were, then, observed. These include entertainment, culture, tourism, education, interactive data visualization and crisis management. After that, one important facet of VR development was discussed, the health issues it has the potential to cause, particularly simulator sickness. In order to understand it better, definitions of presence,immersion and latency were presented, as these factors are the major contributors to simulator sickness. Finally, strategies to minimize discomfort in a virtual experiences were presented in the form of design guidelines.

[11] VR Design Best Practices Oculus' guidelines for VR development. - Last accessed: 28/07/2020

[12] Designing for Google Cardboard Google's guidelines for VR development. - Last accessed: 28/07/2020

[13] Cardboard Design Lab Google's mobile app exemplifying their VR design guidelines. - Last accessed: 28/07/2020

Chapter 3

Analysis and System Design

This chapter will elaborate on how the Research Questions mentioned in Section 1.2 will be answered. Starting by stating the necessary features and how they will contribute to the answers. Then, the georeferenced data will be described and following it will be a description of the tools available to the users. Concluding the chapter will be a description of the system's elements and how they interact with each other.

3.1 Requirements

The main research question, defined in Section 1.2, aims to explore how a crisis management scenario can be improved in a collaborative Virtual Reality (VR) environment. From this overarching question a several requirements can already be identified. The users will need to be in a shared Virtual Environment (VE), facing a challenge that must be analysed, the crisis scenario, a wildfire that was previously recorded. An interface should be available to interact with and visualize georeferenced data on. Users must have meaningful ways to interact with the environment around them and communicate amongst themselves. Thus, the system must import, process and display geographical data, and the tools must have geographical significance.

Summarizing, the system must implement these necessary features:

- A VE supporting the presence of *multiple users* simultaneously connected *over the network*;
- A *wildfire playback*, for the users to analyse the *challenge*;
- *Programmer tools to import geographical data* from the affected region;
- User tools to interact with the VE and communicate with other users;

- User tools to interact with one or several maps.

Other desirable characteristic is the modularity of the system: user tools should be developed independently from other tools and their functionality self-contained; geographical data should be changeable and the system able to adapt to a new dataset, provided it is correctly preprocessed. To help in the implementation of the system a case-study was devised.

Case-Study The case-study was based around the municipality of Mação, a rural territory that suffered from several wildfires over the course of many summers. The wildfire this system is focused on happened during the month of July in 2019, for which the Gabinete Florestal de Mação provided the unprocessed data. The Centro de Informação Geoespacial do Exército[1] provided the high-resolution heightmap data used to generate a tri-dimensional view of the region. Remaining data (related to roads, waterways, etc.) was sourced from Open Street Map (OSM)[2] (© OpenStreetMap contributors). The case-study focused on having multiple users, an expert and an apprentice for example, in VR sharing the same VE while visualizing and interacting with each other and the georeferenced data. To achieve this goal, a set of VR interactions were designed that allow users to; visualize the affected area and the wildfire progression over time, spatially and temporally annotate the terrain, command vehicles, measure distances, identify towns, visualize how different wind directions could affect wildfire progression, draw on the terrain, and notify other users of new information in the system. Some of these interactions are made through the interface itself and some others through specialized VR tools.

3.2 Georeferenced Data Gathering

The data collected consists of publicly available georeferenced data, provided by OSM (© OpenStreetMap contributors) such as geometry representing roadways, waterways, boroughs and towns. A high-resolution heightmap of Mação region, provided by the Centro de Informação Geoespacial do Exército. The elevation data analysed and processed by José A. Gonçalves [16], visualized in Figure 3.1, was also used initially as support data.

A sample of the data used can be seen in Figure 3.2.

One tool implementation needed access to more data, particularly the terrain aspect[3] and terrain slope[4], these datasets were generated from the original heightmaps using the geographic information software QGis[5].

[1] Centro de Informação Geoespacial do Exército - Last accessed: 04/05/2021
[2] Open Street Map - Last accessed: 04/05/2021
[3] The Terrain Aspect is a terrain characteristic that indicates the compass direction the terrain faces.
[4] The Terrain Slope is a terrain characteristic that indicates the inclination the terrain makes with the horizontal.
[5] QGis - Last accessed: 04/05/2021

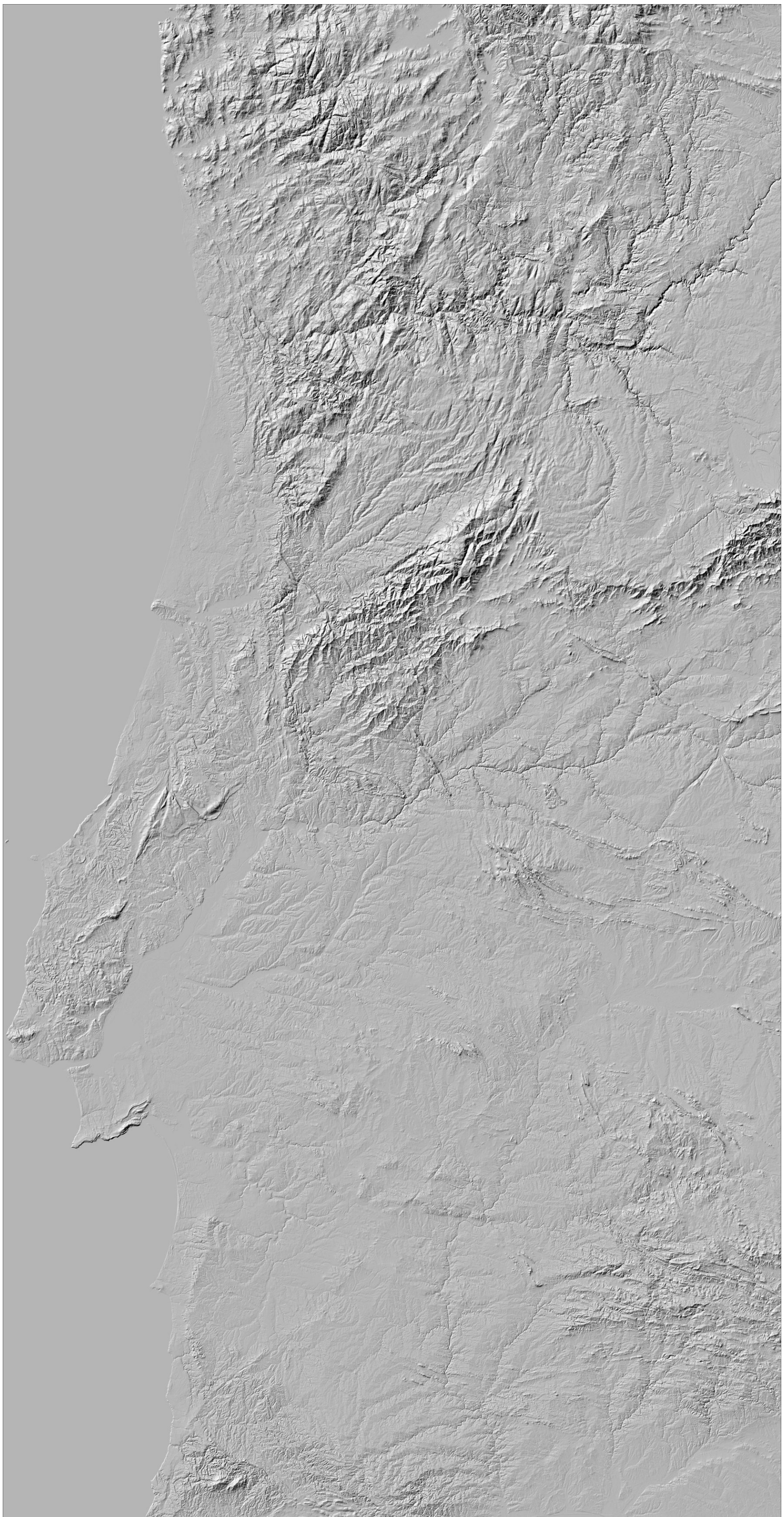


Figure 3.1: Portugal shaded heightmap with vertical exaggeration of 2x [16].

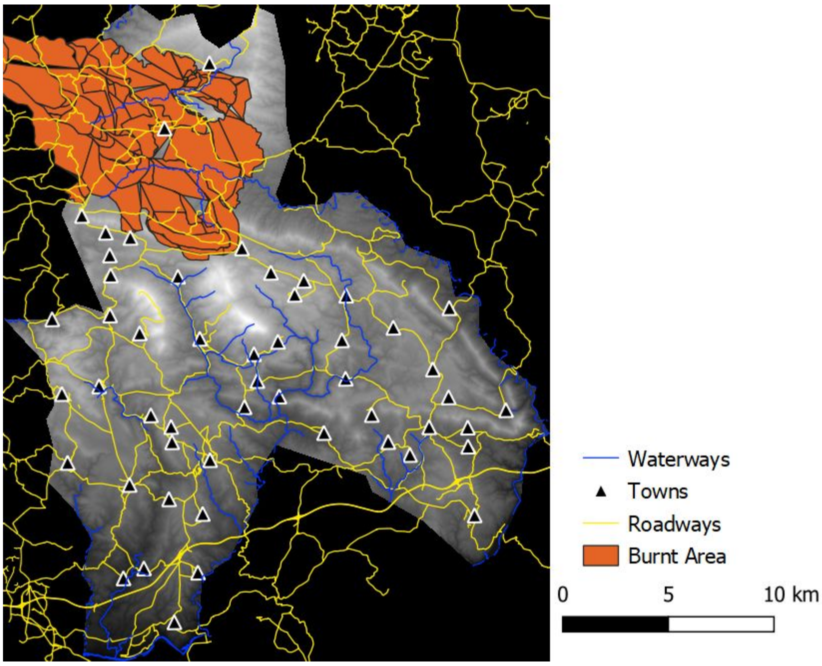


Figure 3.2: Sample of the georeferenced data exported from QGis, with waterways, roads, towns and the total burnt area.

The wildfire data comes from the July 2019 fire, provided by Gabinete Florestal de Mação, that raged through the region and was then recorded, as such, it was not simulated and may have discontinuities.

Most of the data available underwent some processing, to comply with some engine limitations, filter out irrelevant data or just to reproject everything to the same Coordinate Reference System (CRS). These operations will be detailed in Section 4.1.

3.3 Feature Design

Taking into consideration the requirements analysis, there was a feature design phase were several features were discussed and idealized. These features are introduced below and their implementation described in detail over the course of Chapter 4. The designed features are resumed in Table 3.1 along with some that, for one reason or another, were not realized.

3.3.1 Multi-User VE

The main idea for the system is to have a room where multiple users can connect to and be present, while visualizing and extracting information from the crisis scenario.

Multiple Users Multiple users should be able to share the VE and visualize each other. An avatar should fulfil this role, representing user's head and hands, allowing users to perceive what each other are doing and where they are looking at.

Table 3.1: Overview of the designed features and some others designed but not implemented.

Multi-User VE	**VR Tools**	**Supporting Features**	**Not Implemented**
Multiple Users	Marker Tool	Layers	Risk Analysis Tool
Dual-Map Setup	Brush Tool	Notification System	Firefighting Tool
Wildfire Playback	Vehicle Tool	Geometry Importer	Terrain Manipulation
Supporting Interfaces	Ruler Tool	-	Data Filtering
-	Wind Tool	-	-

Dual-Map Setup The VE centerpiece is the dual-map setup. It is an approach that presents the user with two different views over the same data, one is two-dimensional and the other three-dimensional, the former should be placed on a wall, opposite the user, and the latter in a table, centered in the room. Both maps should respond to interactions coming from the users and mirror each other's interactions, meaning that a user can, for example, measure distances in both maps, as seen in Figure 3.3. This dual-map approach is one proposal of this thesis, as its application to geographical data visualization is uncommon, it also allows users the freedom to choose which interface they find most adequate for a given task, while allowing two users to each use one map, sharing information, without interrupting each other.

Wildfire Playback The users should be able to clearly understand how to wildfire progressed. For that to happen, the users should be able to scrub the wildfire timeline, from here on referred to as *Simulation Time*, as if it were a common video player and observe the wildfire progress or regress accordingly and in real-time. The wildfire should also visually reflect the state of its firefronts, in order to discern, for example, high activity fronts from extinct or low activity ones. This feature would add a temporal component to the system which some tools could take advantage of.

Supporting Interfaces Some features or tools might expose adjustable parameters that the user could interact with, although infrequently. These parameters should be placed opposite the main interface, behind the users initial position as to not obstruct the main dual-map interface.

3.3.2 VR Tools

Several tools were devised to be available in the VE. Some should be used to extract information from the map while others should be more focused on communicating some aspect to other users.

Five tools made the final version of the system and their design is presented below. Some tools were designed but were dropped during development for one reason or another.

Tool Wheel With five tools available a mechanism for selecting them had to be designed. The idea is to have one hand always be the tool and the other hand always be the tool selection. This way, the user can interpret as one hand being the active one and the other supportive. The tool selection should be similar to a radial menu, presenting the user with all available tools at once but having the highlighted tool display by its name and description.

Marker Tool The *Marker Tool* should allow for users to point to a location and tag it, leaving a marker at that location with a pending request, such as "Evacuate Civilians", as illustrated in Figure 3.4. These markers could take advantage of the temporal component defined before in the *Wildfire Playback* paragraph. Allowing them to exist both spatially and temporally, reacting to the user scrubbing the *Simulation Time*.

Brush Tool The goal of the *Brush Tool* is to allow users to freehand draw on the maps, allowing for a more natural expression of intent while communicating with other users. Arrows, perimeter lines and other types of scribbles should be easy draw using this tool. It should also expose some customization options to the user such as color and size.

Vehicle Tool The dual-map setup should have representation of resources in the form of vehicles. These vehicles should allow for a user to dispatch them to any location on the map, while indicating its planned route. This is another tool that should take advantage of the temporal component present in the system, moving the vehicles congruently with the *Simulation Time*.

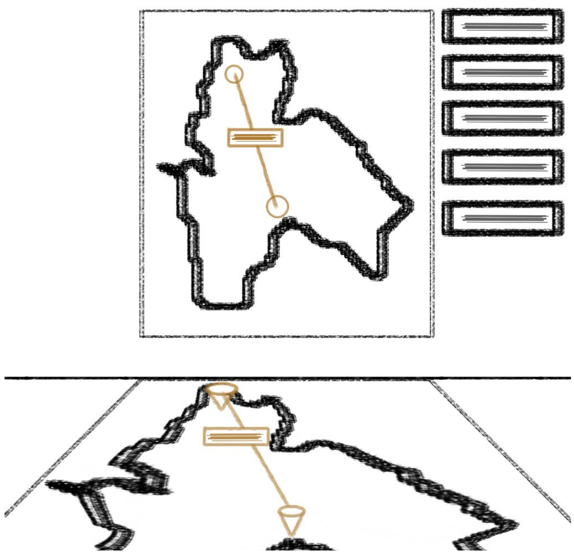


Figure 3.3: Example of the measuring tool showing how both maps mirror each other.

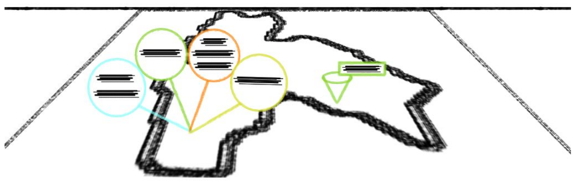


Figure 3.4: Example of the marker picker on the left side of the sketch and a placed marker on the right side.

Ruler Tool In most cases when maps are involved at some point there will be a need to know the distance between points. Then, users should be able to measure distances between arbitrary points in the map. The *Ruler Tool* could help with this, allowing users to place multiple points on the map and see the distances between each point.

Wind Tool During a wildfire sometimes it is necessary to understand how the terrain topology combined with the wind can affect the fire propagation. To make that understanding easier, the *Wind Tool* was designed, it should allow users to control wind direction and visualize how much it will affect the terrain topology, seeing which slopes are harsher and more wind exposed.

Tools Designed but not Implemented Other tools were designed but their implementation was not realized. The *Risk Analysis Tool* was designed to warn the users of towns that were in imminent danger by virtue of being close to the advancing wildfire. Another tool designed was the *Firefighting Tool* that allowed the commanded vehicles to combat the wildfire, this was ultimately decided against as it would add considerable complexity to the system.

3.3.3 Supporting Features

Some other features were designed to support the case-study, there should be support for multiple layers to be visualized simultaneously, as well as a mechanism for users to inform others of new information.

Layers The system should support various kinds of layers to show different types of information simultaneously and having the ability to select independently which should be on or off. The main layers should be; waterways, roadways, boroughs, towns, forests and wind. These layers are the most relevant for the case-study here explored.

Notification System Some tools should have the ability to notify other users of their action. The users would be alerted visually and through haptic impulses on their controllers. Figure 3.5 illustrates the visual component of one type of notification. Complementing

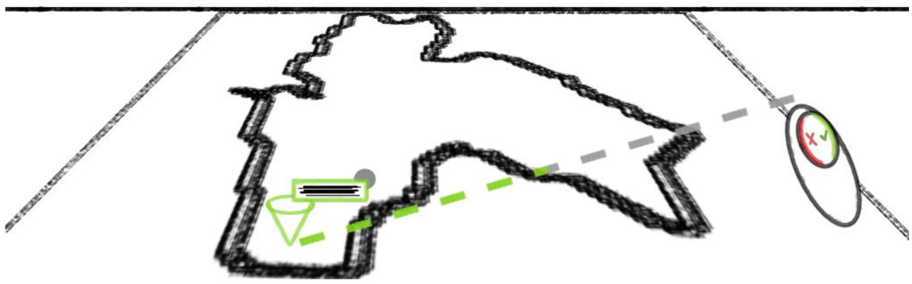


Figure 3.5: Example of one notification type, connecting both controller and marker. The controller also allows for approving or refusing the marker proposal.

this notification system, there should be a way for users to propose, accept and decline actions executed by the tools.

Geometry Importer A necessary programmer tool was developed to assist pre-visualizing and importing the geolocated data exported from QGis. Aptly named *Layer Creator*, this Unity editor tool allows the programmer to import *GEOJson* files that contain polylinear, polygonal or punctual data and create both a texture and a game object that represents that data, particularly useful for creating the three-dimensional town representation. In Figure 3.6 we can see the tool in use to import several types of data. How the tool creates the textures is detailed in Section 4.4.1.

Features Designed but not Implemented One interesting feature that was planned was to allow the users the ability to *manipulate the terrain's rotation and scale* and be able to fly through the landscape. But due to the constraints the engine puts on the terrain

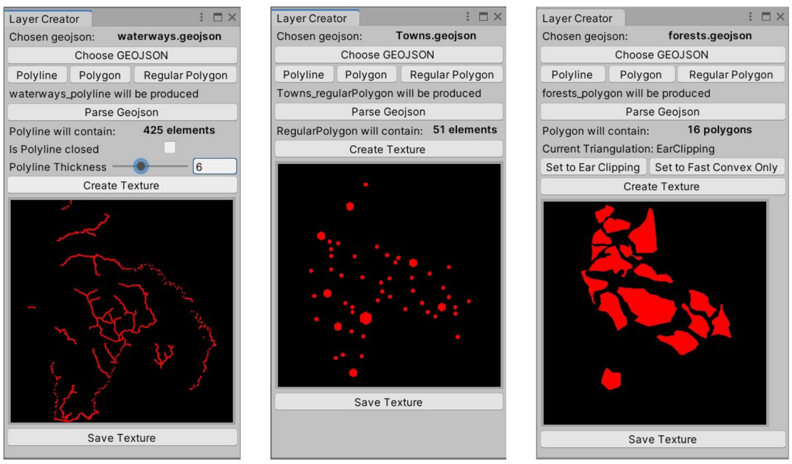


Figure 3.6: The Layer Creator tool. On the left, the programmer is importing 425 polylines representing water courses. The center image represents the 51 towns in the region. On the right, the tool parsed the 16 polygons representing forest patches.

object it was impossible to rotate and very costly to rescale, so this idea was ultimately decided against. Another feature that was design was the ability to *filter the georeferenced data* based on an area designed by the user, this feature was also left unimplemented.

3.4 Technologies and Architecture

The target platform for this thesis was the HTC Vive VR headset; to develop for it the most straightforward way was to use a game engine. The two main competitors in this space are the Unreal Engine and the Unity Engine, both engines are capable of targeting the HTC Vive and both were good choices for this system. The chosen engine ended up being Unity due to the author having previous experience with it. Thus, the project began as a Unity 2019 game, preferring the Universal Rendering Pipeline over the other two (High-Definition Rendering Pipeline and Built-in Render Pipeline) as it promised better performance over the other two[6].

At the time of starting this thesis, to target the HTC Vive, a developer had to target the SteamVR runtime. Valve provided two ways of doing this, one could use the OpenVR SDK, deprecated as of Unity 2019 or the OpenXR SDK which was in beta.[7]

SteamVR - OpenXR First test was using the OpenXR beta. The setup was non-standard, being in beta the plugin was not available directly from Unity's Asset Store and had to be installed through a standalone package. That was not a problem as it was a one time operation. Unfortunately, the plugin was not stable enough to develop with, every once in a while when entering Play Mode the editor would crash with no feedback and upon a brief, and unfruitful, investigation OpenXR was abandoned.

SteamVR - OpenVR Then OpenVR was tested. Installation process was virtually non-existent has OpenVR had built-in support into Unity 2019 even though it was deprecated. Everything worked as expected and there weren't any relevant details to using OpenVR, which ended up being the choice for this system.

VR toolkits are also helpful as they implement common VR functionalities and are essential for a system of this scale; small demo with only one programmer. When starting development three popular toolkits were available.

SteamVR Interaction System Valve published a demo made in Unity showcasing their interaction system for VR, it was a comprehensive demo showing the multiple capabilities

[6] This ended up not being a very good choice when it came to writing the shader code for the visualizations; the Universal Rendering Pipeline does not support hand-written shaders, they must be authored using a node-based language that at time of implementing did not support terrain shaders. The route taken was to hack away at Unity's internal terrain shaders, which did the job but was unnecessarily complex. In hindsight, the Built-in Render Pipeline should have been chosen.

[7] At time of writing and with the release of SteamVR 1.16, the integration with OpenXR has reached 1.0.

supported. It is primarily aimed at developing videogames. The system showed no cons besides not being built-in into Unity.

Virtual Reality Toolkit VRTK[8] is an open source toolkit aimed at VR developers. Supporting much of the same features as the SteamVR Interaction System. At the time of starting the thesis there were some compatibility issues and SteamVR 2 was not officially supported.

Unity XR Toolkit The chosen toolkit ended up being Unity's own toolkit that was in preview when development started, even though it had much less functionality then the other two, it had better integration with the editor and had all the functionality necessary for this particular use case.

Geographical Processing Some facets of development required processing of geographical and image data. To fill those roles, two softwares were used; Krita[9] and QGIS[10] both chosen for being Open Source Software. Their uses will be detailed in Section 4.1.

Networking As a networking layer there were two main choices, Photon PUN[11] and Mirror[12]. Both these technologies were tested and the results are detailed in Section 4.2. Due to limitations present in the lab's network environment the Open Source Mirror network layer was favored over the proprietary Photon PUN.

Remaining tools, assets and other resources from third parties used throughout this thesis are listed in Annex III.

Diagram 3.7 illustrates the main modules present in the system. On the left, is presented the *Main Client* denominated as such because this is the client that also acts as a server, other clients connect to this one, it also hosts the Open Source Routing Machine (OSRM) server that allows vehicles to plan their routes. On the right, there is another client connecting to the main one. Connecting the two is responsibility of the system's *Network* layer. The main modules that are present in all clients are the *Georeferenced Data* that contains all the pre-processed georeferenced data, including wildfire data, the *VR Interface* module, allowing users to interact with the environment and *Rendering* the module that allows the users to see everything they see in the VE. These modules will be the focus of the next chapter, Chapter 4.

To implement the features in the case-study scenario, the architecture needs to support VR interaction and collaboration. Exploring collaboration in VR requires networking. In order for all the previous mentioned modules to behave correctly on multiple clients a well thought out architecture must be well thought out.

[8] Virtual Reality Toolkit - Last accessed: 04/05/2021
[9] Krita - Last accessed: 04/05/2021
[10] QGis - Last accessed: 04/05/2021
[11] Photon PUN - Last accessed: 04/05/2021
[12] Mirror Networking - Last accessed: 04/05/2021

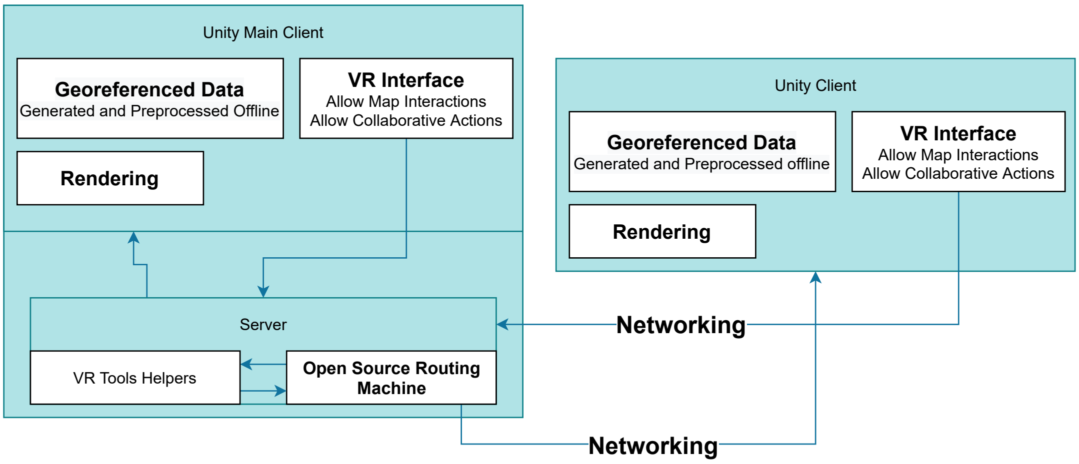


Figure 3.7: High-level overview of system components.

In order to separate the networking aspect from the local aspect of the *VR Tools* a modular approach was taken. Each tool should have two components one *User Component* and one *System Component*. The former should be responsible for input and local interaction while the latter should translate the intent and replicate it to the network. This approach should decouple the local actions from the networked actions and result in a more modular approach to tool development.

Since implementing low-level networking features was not the goal, a networking library was used, Mirror, and an uncomplicated architecture followed. The system uses a client-server architecture where one user acts as both client and server and the remaining users act as clients connecting this server. In order for the tools' impact on the VE to be seen by all users their actions are always sent to the server and then the server sends their results to all clients. This was a model inherited from Mirror, the focus of this thesis was not in the improvement of the network protocol.

Diagrams 3.8 and 3.9 illustrate the architecture followed, showing how both the local and remote clients ask the server to make actions and then it orders the execution on both.

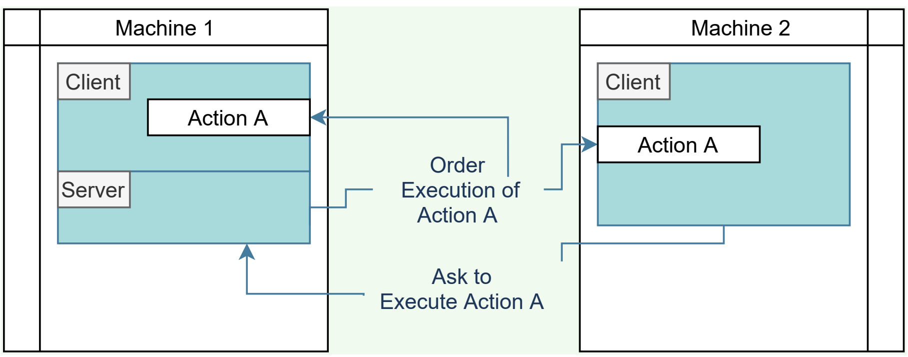


Figure 3.8: Diagram illustrating how the *remote* client executes an action on both VEs.

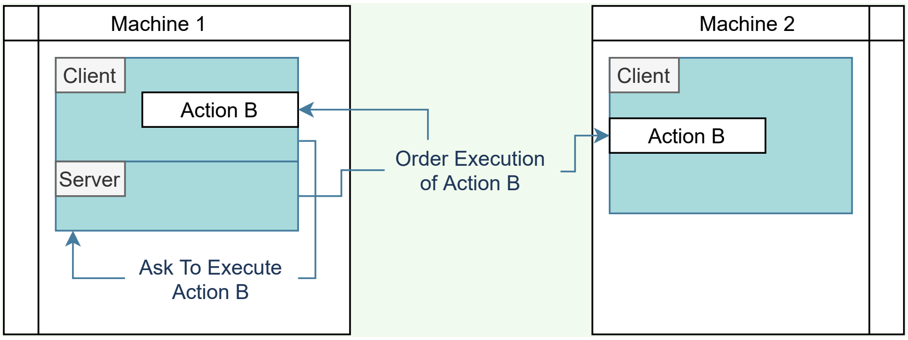


Figure 3.9: Diagram illustrating how the *local* client executes an action on both VEs.

3.5 Summary

This chapter detailed the elements needed to answer the proposed research questions; it specified the requirements the system must abide by, introduced the data that will be analysed and catalogued and the tools the users will be handling to interact with the VR and communicate with each other. Concluding this chapter, some secondary but necessary features were mentioned and the architecture of the whole system presented.

Chapter 4

Implementation

This chapter will detail the implementation of the Virtual Reality (VR) collaboration system using the case study described at the end of Section 3.1. Using Diagram 4.1 as a guide, already mentioned in Chapter 3, this chapter will first look at the *Georeferenced Data* module in Section 4.1; the types of data the system is able to visualize, how this data was sourced and what transformations were required to facilitated the importing process. Then the *Networking* module will be explained in Section 4.2, even though networking was not the focus of this thesis it took a considerable amount of effort to get the correct behavior across clients so explaining how it was leveraged in this particular system will be relevant. Then the *VR Interface*, in Section 4.3, will detail which tools are available for the user to interact with as well as other relevant interface elements. Then the *Rendering* module will be explained in Section 4.4, going over how the system implements the multiple visualizations available Following and concluding the chapter, the collaborative actions users can take within the Virtual Environment (VE) will be detailed in Section 4.5.

4.1 Georeferenced Data

The test case for the system centers around a wildfire that burned the region of Mação for almost a week. So having geographical data at user's disposal is fundamental. In order to display this data within Unity some steps had to be taken in order to source, process, import and visualize it. The system deals with height data in the form of high-resolution heightmaps to generate the three-dimensional map, and geometry data in the form of points, lines and polygons to represent towns, roads and forest patches respectively. Data related to the wildfire can be categorized as geometry, seeing as it consists of lines denoting the current fire fronts and polygonal data denoting the burnt area.

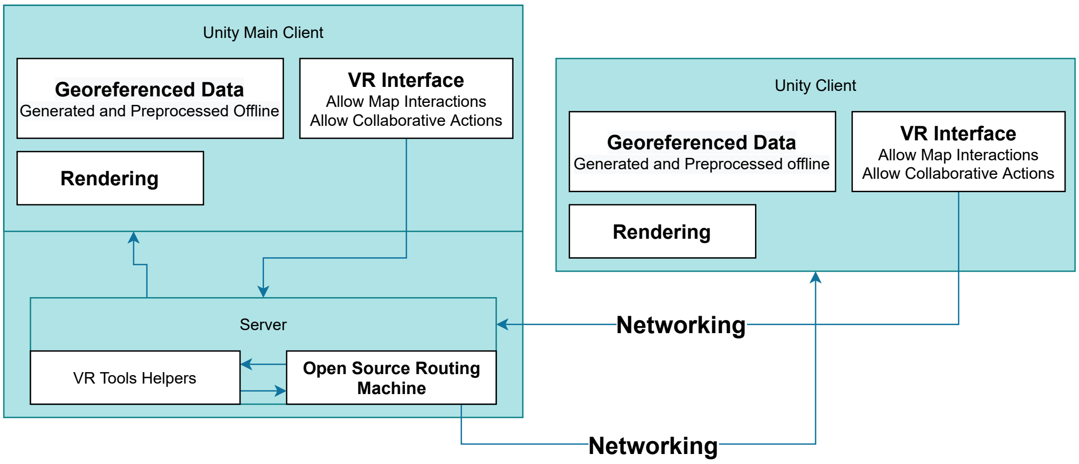


Figure 4.1: High-level overview of system components.

Also related to georeferenced data is the integration with the Open Source Routing Machine (OSRM) that supports the *Vehicle Tool* with routes for the vehicles.

4.1.1 Terrain Height Data

To create the three-dimensional map of Mação, the Unity Terrain System[1] was used. The map was constructed based on a high-resolution regional heightmap of Mação, Figure 4.2, in a *sixteen bit unsigned integer format*, provided by the Centro de Informação Geoespacial do Exército[2]. With 6664x8017 pixels, the map covers the entire region of Mação municipality with the outside of it having no data. Each pixel covers an area of 5x5 = 25 squared meters; the pixel values represent height of the terrain and have a minimum value of 0 meters and maximum of 630 meters.

Unity's terrain system imposes some constraints on the heightmap, it must have a resolution that is a power of two plus one, (513x513, 1025x1025, ...) up to a maximum of 4097x4097. Thus, the original heightmap must be transformed to fit within this limitation. Using the open-source image editor Krita, a border around the original image was added with zeroed data in order to make it the nearest higher power of two, 8192x8192. Then the image is resized, using bicubic sampling, to the maximum resolution supported by Unity's terrain system, 4097x4097. At this point, Unity can import the file, provided Krita exported it as a *R16 Heightmap (.r16)* and its extension is then changed to *.raw* in order for Unity to recognized it as suitable for the terrain system. Although, if imported as such, the terrain will reflect the noise present in the image, as seen in Figure 4.3. To avoid this, a Gaussian blur with a radius of five pixels is applied yielding a smoother terrain without the spikes.

In order to have georeferenced data placed correctly on the terrain, its Coordinate Reference System (CRS) must be taken into account, all data must be reprojected to

[1] Unity Terrain System Documentation - Last accessed: 04/05/2021
[2] Centro de Informação Geoespacial do Exército - Last accessed: 04/05/2021

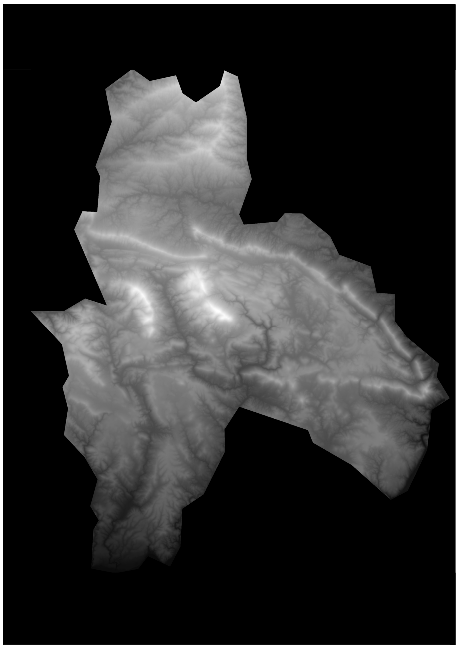


Figure 4.2: Heightmap of the Mação region.

that same CRS. Then with the four corners of the terrain extents being known as a geographical coordinate the necessary conversions to the cartesian coordinates can be computed and everything falls in its place. The CRS used was the *WGS 84 / Pseudo-Mercator EPSG:3857*. Regarding the scale of the maps, the region represented in the VE spans a square region of 40 kilometers on each side, covering a total area of 1600 squared kilometers. Some of this area has no data as it lies outside of the Mação municipality. These 40 kilometers, 40000 meters, are represented by a Unity 3D terrain with edges measuring 1 Unity unit which correspond to 1 real-life meter. Meaning the 3D terrain is at a scale of 1:40000. The 2D map has 2 Unity units, 2 real-life meters, on each side, representing the Mação region at a 1:20000 scale. The 625 meters, which are the peak

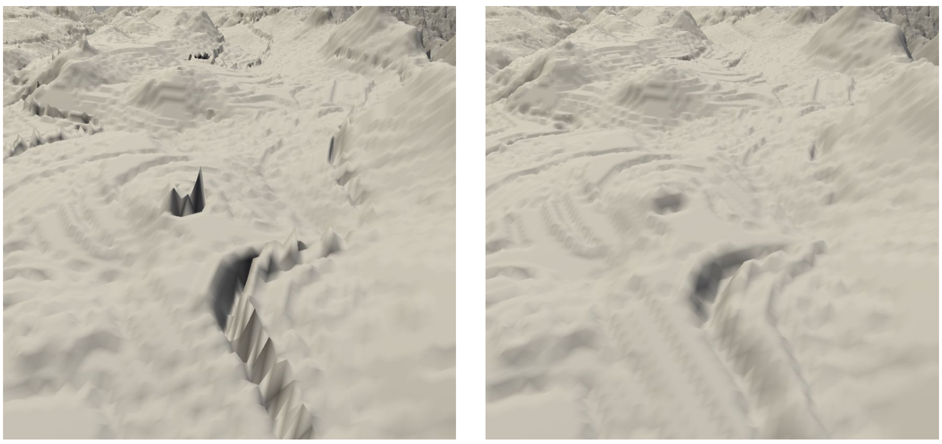


Figure 4.3: Left: Terrain imported without a denoising step; Right: Terrain after a Gaussian blur applied.

Table 4.1: Overview of terrain scales. The region represented is a square region, as such its *Width* is the same for both sides. *Height* refers to the height of the terrain itself.

Inter.	Dimension	Real Dimension	Unity Dimension	Scale
-	Base (meter)	1 meter (m)	1 Unity unit (Uu)	1:1 m
-	Base (kilometer)	1 kilometer (km)	1 000 Uu	1:1 000 m
Wall	Region Width	40 000 m	1 Uu	1:40 000 m
Wall	Region Area	1 600 km^2	1 Uu^2	1:1600 km^2
Table	Region Width	40 000 m	2 Uu	1:20 000 m
Table	Region Area	1 600 km^2	2 Uu^2	1:800 km^2
Table	Terrain Height	625 m	0.1 Uu	1:6250 m

height in Mação, are represented as 0.1 Unity units, corresponding to 0.1 real meters, meaning the height is at a 1:6250 scale. Table 4.1 resumes this information.

4.1.2 Geometries - Points, Lines and Polygons

Another component relevant for the system is the ability to analyse layers with different information projected on the terrain. These layers come in the form of points, lines and polygons representing towns, water courses and roads, and patches of forests. The data for them was sourced through Open Street Map (OSM)[3] and imported to QGis[4]. The relevant data pertains to the water courses, roads, boroughs, towns and forests.

The data downloaded from OSM suffered some transformations in QGis; it was first reprojected to the same CRS as the heightmap, then because it covered an area larger then necessary, was *clipped by the heightmap's extents*. Following that, filtering was applied in order to exclude some less relevant features. One particularity that must be mentioned, is that the OSM had very little information regarding forest patches, so these were hand-made (consulting satellite photography) in QGis and do not necessarily represent real forest patches.

Once the data was processed, it was exported as *.geojson* files and imported into Unity. This importer, aptly named *Layer Creator*, is a custom made tool to import these features and is as generic as it needed to be for this particular case. The importing process has a couple steps to it; first the user chooses the file to import, following that they choose how that data is supposed to be displayed (as Polylines, Polygons or Regular Polygons for punctual data), after that the tools parses the file and informs the users of the number of elements processed and some configuration options, then the user can preview the texture that will be generated, save it to disk as well as saving a Unity game object to be used later, this last step is of particular importance to the layer representing the towns in 3D. Figure 4.4 illustrates each type of geometry present; points, lines and polygons and Section 4.4 details how the steps necessary to visualize it.

[3] Open Street Map - Last accessed: 04/05/2021
[4] QGis - Last accessed: 04/05/2021

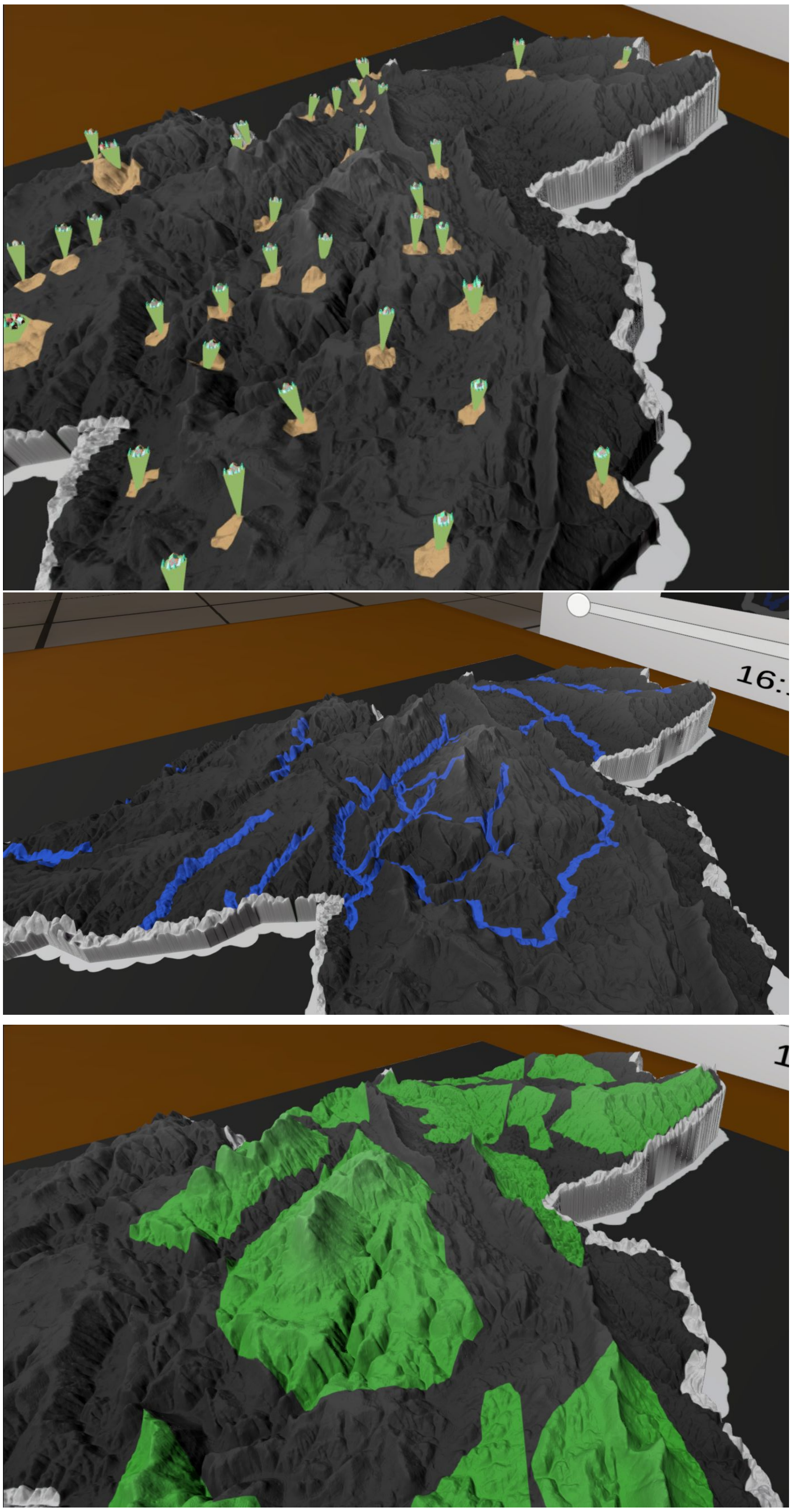


Figure 4.4: Example of the different types of georeferenced geometry supported. Top image, represents points (towns), middle image, lines (waterways), and bottom image represents areas (forests).

4.1.3 Wildfire Data

The wildfire's representation had a similar treatment to the other kinds of data (importing to QGis, reprojecting, exporting as *.geojson*), as it is composed of lines denoting the fire fronts and polygons denoting the burnt area. Some notable differences were the data related to start and end times of fire fronts and burnt areas, as well as intensity values for the fronts. This data is, of course, preserved and used later in Unity. As the fire is a dynamic element present in the VE, the data is not imported through the *Layer Creator* custom tool. It is, instead, loaded at the start of the application, constructing multiple game objects that represent each element present in the data. These game objects are not rendered to the main cameras, but are instead rendered to an offscreen renderbuffer, the reason why can be read in Section 4.4. Their positions are also used as spawn position for several particle systems. The fire burning can be seen in Figure 4.5.

4.1.4 Wind Layer

Some relevant information to have in a wildfire scenario is how the wind can impact the terrain, helping or deterring flame progression. With that in mind, a visualization of the affected terrain was implemented as the *Wind Layer*. This layer shows how the terrain is affected by the wind's blowing direction and an example can be seen in Figure 4.6. How the visualization is achieved is detailed in Section 4.4, here it will be mention what data is needed and how it was generated.

For this visualization to be computed it is necessary to have some more data about the terrain, other than its height. Two terrain characteristics are needed to be exact, the *slope* and its orientation, the *aspect*. This data was generated from the original heightmap using QGis. If the visualization would be only present in the 3D map, there would be no

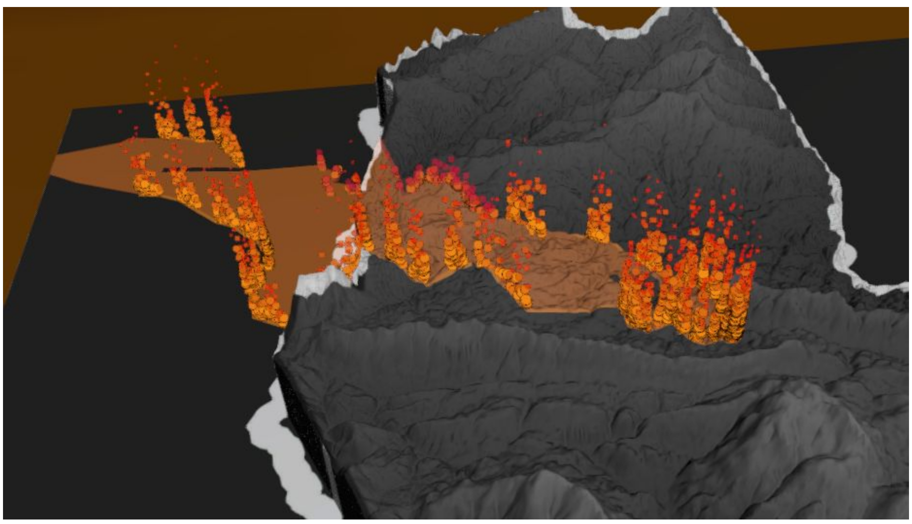


Figure 4.5: Wildfire burning closeup. Particles emitted from active fire fronts have different characteristics (frequency, color, lifetime) according to the front's activity.

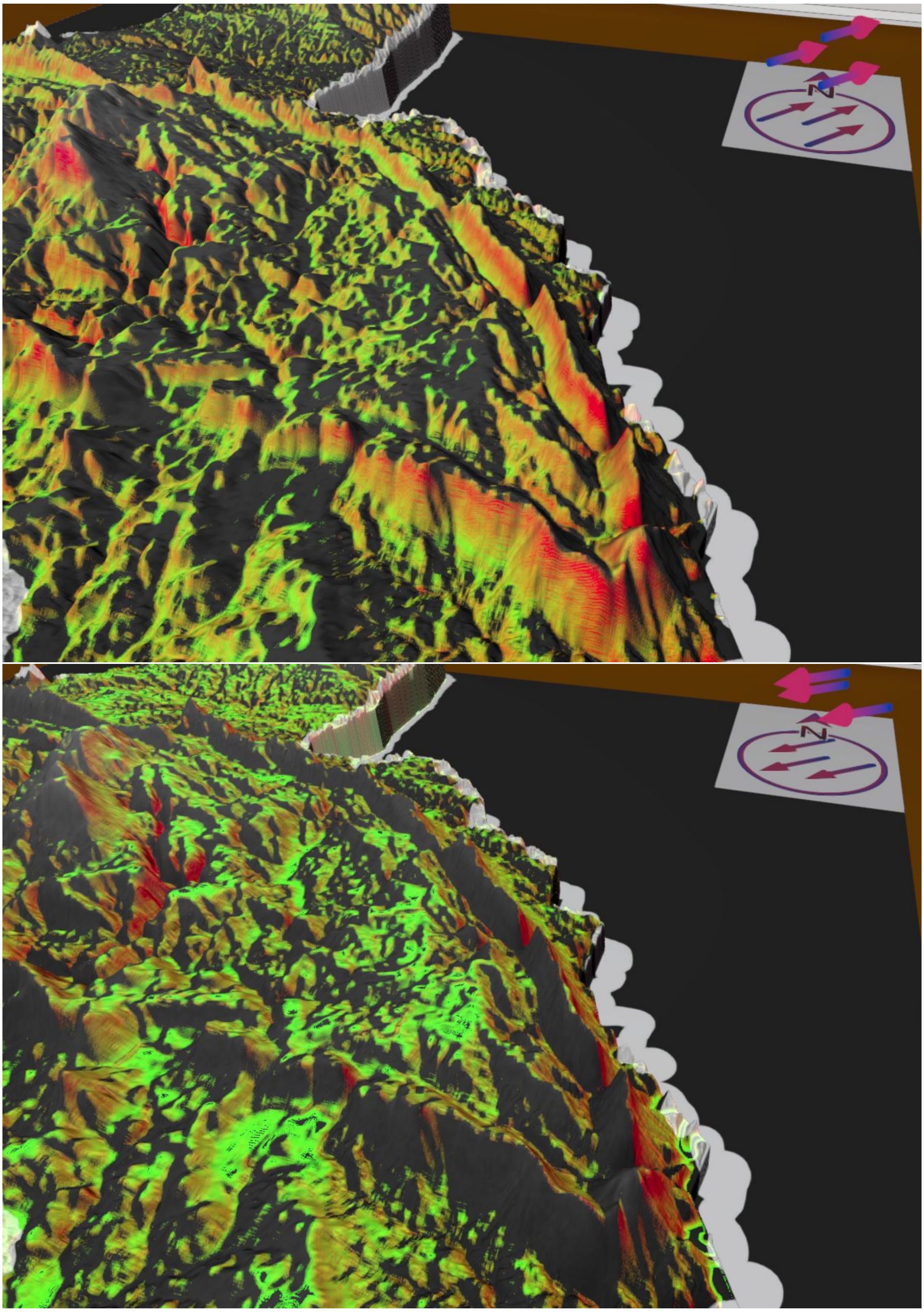


Figure 4.6: Top: Wind blowing from Southwest to Northeast, affecting mainly the higher slopes oriented towards Southwest. Bottom: Wind blowing form Northeast to Southeast, the same slopes are now shielded from the wind.

need to generate this information; the terrain normal already generated by Unity could give that information.

Slope The *slope* value of a given patch of terrain indicates the steepness of that patch with regards to the horizon, in a 0 to 90 degree range. For example, a completely flat patch of grass would have a *slope* closer to zero, whereas a location close to the peak of a mountain would have a value much higher. On the left side of Figure 4.7 we can observe the *slope* generated using GDAL's[5] *Slope* algorithm over the original heightmap terrain, in Figure 4.2, where the darker values represent a fairly plain slope, brighter values represent steeper slopes and the dark values outside the region's limits have no data.

Aspect The *aspect* value of a given slope on the terrain indicates which cardinal direction that particular slope faces. The value can range from 0 to 360 degrees, with both values representing the slope being due North and 180 degrees being due South. The right side of Figure 4.7 presents a visualization where brightness values close to each other are oriented similarly. This visualization was also generated using GDAL's *aspect* algorithm over the original heightmap terrain, mentioned earlier in Figure 4.2.

4.1.5 Open Source Routing Machine (OSRM) Server

One last aspect still related to georeferenced data is the integration of the OSM [27]. This integration enabled the system to simulate vehicle movement using adequate roads and approximating a plausible speed following the *Simulation Time*. The integration was

[5] GDAL - Last accessed: 04/05/2021

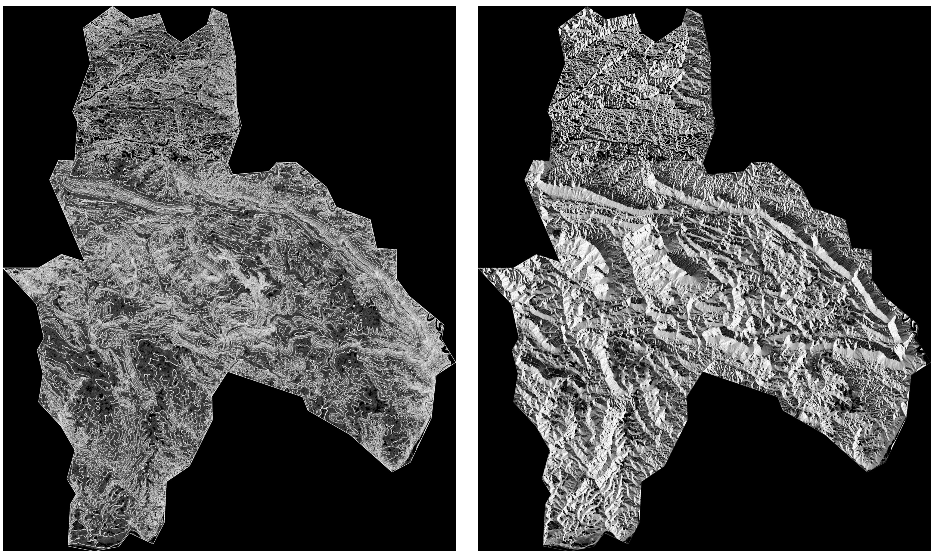


Figure 4.7: Left: Terrain Slope (contrast slightly adjusted for clarity). Right: Terrain Aspect.

simple and is detailed in the Github repository [6]. It does require a server to be running so the path-finding requests can be served. In this thesis, the OSRM server is running on the same machine as the system's host. Then, whenever a user orders a vehicle to move to a position, both the current vehicle position and its destination are sent over to the host that then sends an HTTP request to the server, it responds with the *Route* object with information regarding distance, duration and most importantly the route geometry which is then painted on the terrain. Diagram 4.8 illustrates these steps.

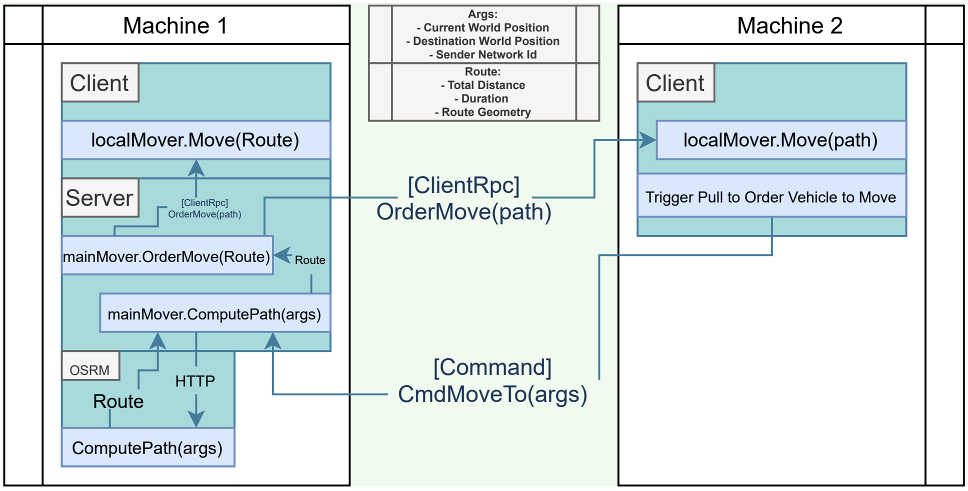


Figure 4.8: Diagram of what happens once a user makes a move order on a vehicle in the VE.

4.2 Networking

Networking features are fundamental to the development of this system. Users must see each other, be able to affect each other's copy of the VE and communicate amongst themselves. In order to augment the system with networking features two different Unity packages were tested. Photon[7] and Mirror[8].

Photon PUN Preliminarily, the Photon PUN package was tested. Documentation along with examples, including a VR example, on the website helped implement small demo quickly with only the *Marker Tool* implemented. The architecture was such that both clients would be connected over the internet to Photon's proprietary servers. This was not seen as a limitation until the point were local testing was to take place. Since testing a VR system requires two headsets it also requires two computers. In the lab, these are network restricted and could not connect to Photon's server. The self-hosted option was briefly explored but quickly abandoned in favour of Mirror.

[6]Open Source Routing Machine - Last accessed: 04/05/2021
[7]Photon PUN - Last accessed: 04/05/2021
[8]Mirror Networking - Last accessed: 04/05/2021

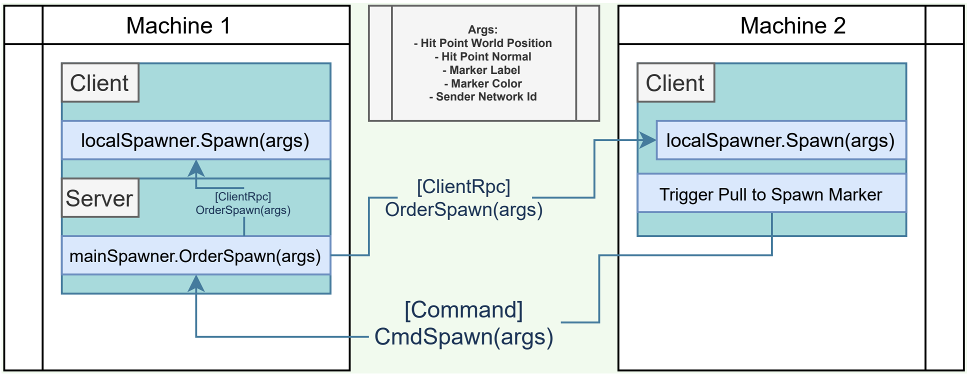


Figure 4.9: Diagram of what happens once a user places a marker on the VE.

Mirror Mirror is an open-source high-level networking API, built as a drop-in replacement for Unity's own, deprecated, networking stack. Once Photon's limitation was clear, work to re-implement the prototype started; Mirror does not provide hosting, so the server would be self-hosted by default, thus enabling local network testing in the lab.

The system must synchronize several things, most importantly the user's transforms and their actions within the system.

Object transform (position, rotation and scale) synchronization is simple and straightforward thanks to the functionality provided by Mirror. Any Unity game object with both a *NetworkIdentity* and a *NetworkTransform* components has its transform automatically synced between all clients. Syncing the tool actions revealed to be not as straightforward, requiring a different approach. To understand this approach, two concepts must be introduced. These are related to how Mirror allows performing actions across the network[9] and are often referred to as Remote Procedure Call (RPC). Mirror defines three types of RPC, *Commands*, *ClientRpc* and *TargetRpc*. The system implementation only makes use of the first two, so *TargetRpc* will be ignored.

Commands *Commands* are special functions invoked by the client to be executed on the server.

ClientRpc *ClientRpc* are special functions invoked by the server to be executed on the client.

All of the tools use these to concepts to communicate actions across all the users. The Diagram 4.9 illustrates how these functions are called when a user activates the *Marker Tool* to place a marker in the VE. The system calls a *Command* on the server, then the main marker spawner orders the local marker spawners, through a *ClientRpc* call, to spawn the markers on their respective environments, culminating in both clients having a marker spawning at the desired position.

[9] Mirror Networking - Remote Actions Documentation - Last accessed: 04/05/2021

4.3 VR Interface

This section will focus on implementation details of the various tools that are part of the system. The tools are similar in how they interact with the network, through *Commands* and *ClientRpc* as mentioned before and shown in Diagram 4.9, but differ in their impact on the VEs. Each subsection below will analyse how the tool works from the user side, how one user does what, and from the system side, how the system manages the actions the tools are capable of making.

But first, an overview of what the interface looks like and the elements present in the dual-map setup, present in Figure 4.10. This state is something the user could encounter after interacting with the system for a little while. It is possible to observe: A - the *Play* button, B - use of the *Brush Tool*, C - the wildfire progress, D - use of the *Ruler Tool* and E - the available layers with *Waterways*, *Roadways*, *Towns* and *Vehicles* turned on. Using the *Simulation Time* on the bottom of the image, it is possible to determine at what time the wildfire was being represented; the dual-map setup represents the wildfire as it was Sunday at 3:19am.

Behind the user, there are two supporting interfaces that expose some parameters. These parameters control the *Brush Tool* and the *Wind Layer* and allow for some customization from the user, they are illustrated in Figure 4.11.

Now onto the tools that are part of the available interface. Each of the below tools is composed of two components; one, referred to as *User Component*, is responsible for

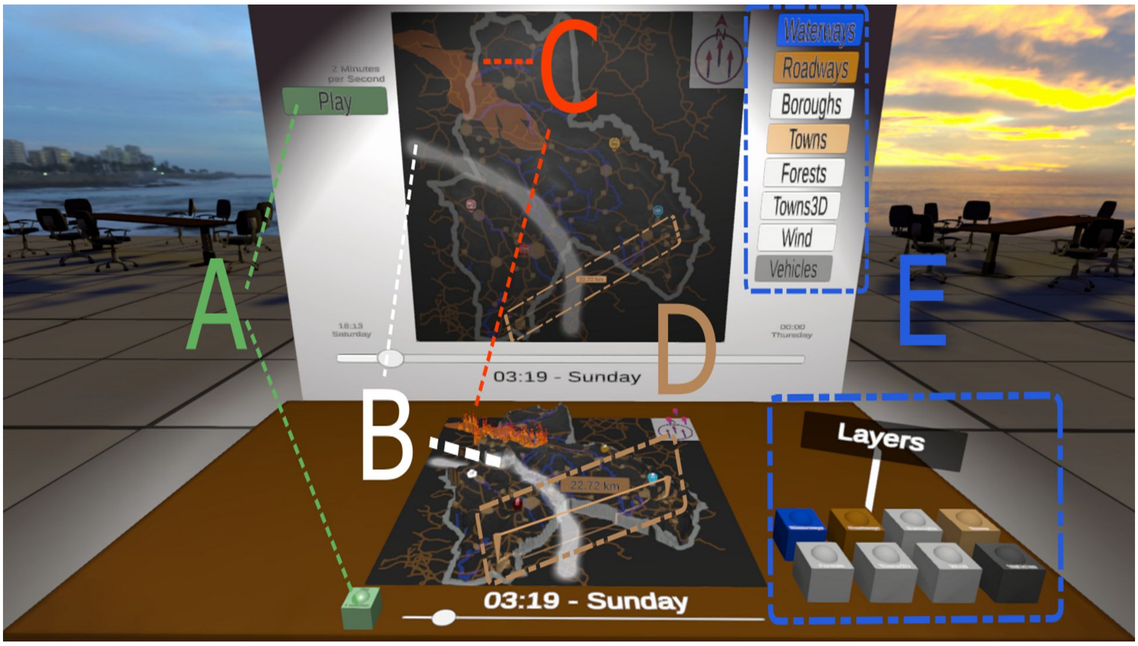


Figure 4.10: View of the dual-map interface in VR. On the left, an overview of some of the features; A - Play button to control wildfire playback; B - Brush strokes from brush tool; C - Wildfire visualization; D - Measurement made with ruler tool; E - Layers with geographic information, activated are Waterways, Roadways, Towns and Vehicles

receiving input and requesting actions the user wants to execute - the other, referred to as *System Component*, is responsible for receiving and synchronizing those actions across the clients. This second component calls *Commands* which are then executed server-side by the other component, which then calls *ClientRpc* on every client, making sure the action is effectively called in all clients. This scheme turned out to be modular enough to separate user input, user action and synchronization logic and was adapted to every tool, thus below the tools will be described according to these two components, *User Component* and *System Component*.

During development the necessity for an alternative input method to the VR controllers was felt. So in order to improve iteration and testing times a supplemental mode was developed that allowed the developer to use conventional first-person controls using a keyboard and mouse. At this point, the *User/System Component* architecture payed off as once one *System Component* was implemented the *User Component* was similar for both VR and first-person[10] implementation, requiring little extra effort to implement. The first-person alternative method will not be detailed.

Tool Wheel Before presenting the tools, the *Tool Wheel* should be mentioned, since it is the way for users to select which tool to use. The users can press a trigger on their left controller and invoke a radial menu containing all the tools available. The user can then press left or right in the controller trackpad to navigate and select a tool; the current highlighted tool is accompanied by its name and a brief description on how to use it. An example of this can be seen in Figure 4.12.

4.3.1 Marker Tool

First tool implemented was the *Marker Tool*. The initial idea for it consisted of directing the remaining users attention to a particular point. It then evolved into being a tool for tagging a position with an action chosen from a pre-determined set of actions.

[10]The virtual-reality and first-person perspectives are semantically identical, the term first-person is used to refer to the common perspective in shooter-type videogames, while virtual-reality perspective refers to someone using an headset.

Figure 4.11: Left: Interface to customize the *Brush Tool*; user can tweak size, softness and color. Right: Interface to affect the *Wind Layer*, allowing to affect some visualization parameters.

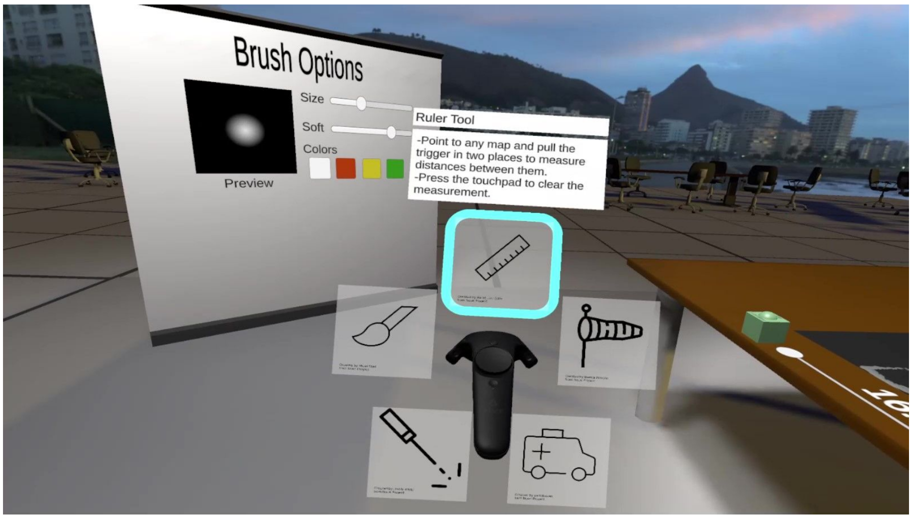


Figure 4.12: Radial Menu - User can see the icon and once they highlight the desired tool, they get a brief description of how to use it.

Usage After selecting the tool the user will point to any place on the map and pull the right trigger, at this point text bubbles will pop-up on the terrain. The user will then point to the desired option and press the trigger again. Once that is done, a new marker will appear with the selected action above it, as seen in Figure 4.13.

User Component The *User Component* is in charge of raycasting to detect if the placement is in a valid object and then showing the action bubbles to be selected. Once the action is selected the component invokes a *Command* to be executed by the *System Component*, passing the relevant data; intersection point and normal from the raycast, the picked action and the user network identifier used by the *Proposal and Notification system*, omitted form the current section and detailed in Section 4.5.2.

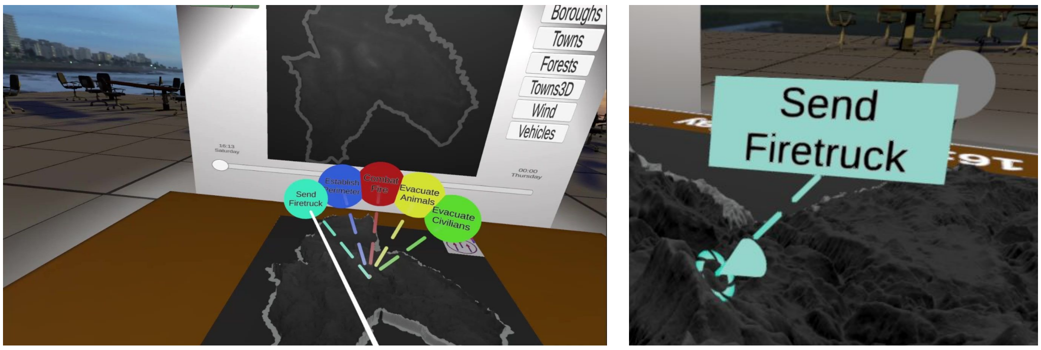


Figure 4.13: Left: User is choosing which marker to place. Right: Close up of the placed marker, with approval pending indicated by the gray circle.

System Component Once the *System Component* receives the *Command* it executes the *ClientRpc* calls ensuring that every client will spawn a marker with the same characteristics at the same position. The *System Component* also keeps track of all the markers spawned. From here on out, the markers are the ones who regulate themselves; upon spawning they record the time they were spawned and the time they should turn off (one hour after spawning), thus being able to turn on and off according to the *Simulation Clock*, no relying on intervention from *System Component* anymore.

4.3.2 Brush Tool

The second tool to be implemented. *The Brush Tool* was initially designed to aid communication between users similar to how scribbling on a piece of paper can supplement verbal communication while conveying ideas. The tool also supports limited costumizability in color, size and softness, as can be seen in Figure 4.14. Network-wise this is the most resource-consuming tool to use.

User Component This tool's *User Component* is similar to the *Marker Tool* one, in the sense that it is only responsible for raycasting into the scene and testing if the hit object is suitable to be painted on. Then, it calls a paint *Command* on the *System Component* while the user pulls the trigger; this is a source of network straining, seeing as every frame the user has the trigger pulled, and is pointing to a surface that is paintable, there are *Commands* and respective *ClientRpcs* being transmitted, the transmitted data itself isn't as lean either; a position, a color, and a some floats describing brush parameters. The local wired testing did not show this to be a problem, but as soon as a wireless laptop was tested these limitations started to show.

System Component The *System Component* is a bit more elaborate as it also deals with the actual texture painting part of the system, detailed in Section 4.4.3 paragraph *Brush Tool*. This component maintains a *Paintable Texture* and directs the painting *Commands* received through the *ClientRpcs* to that texture. This way, each client can keep a texture that has every user's paint strokes with relatively small network traffic, compared to having to synchronize the texture itself.

4.3.3 Vehicle Tool

The idea behind the *Vehicle Tool* was to allow users to have some impact on the scenario, allowing almost direct control over the vehicles and where should they be located to better assist the crisis scenario. The tool allows users to select and order vehicles to move to a destination while correctly following roads. The routing algorithm is provided by OSRM covered in Section 4.1.5. Users must hover their hand over the vehicle to select it and then point and pull the trigger to give it the order to move. Vehicle's movement is tied to the time on the *Simulation Clock*, so if the route computed by the OSRM reports its duration

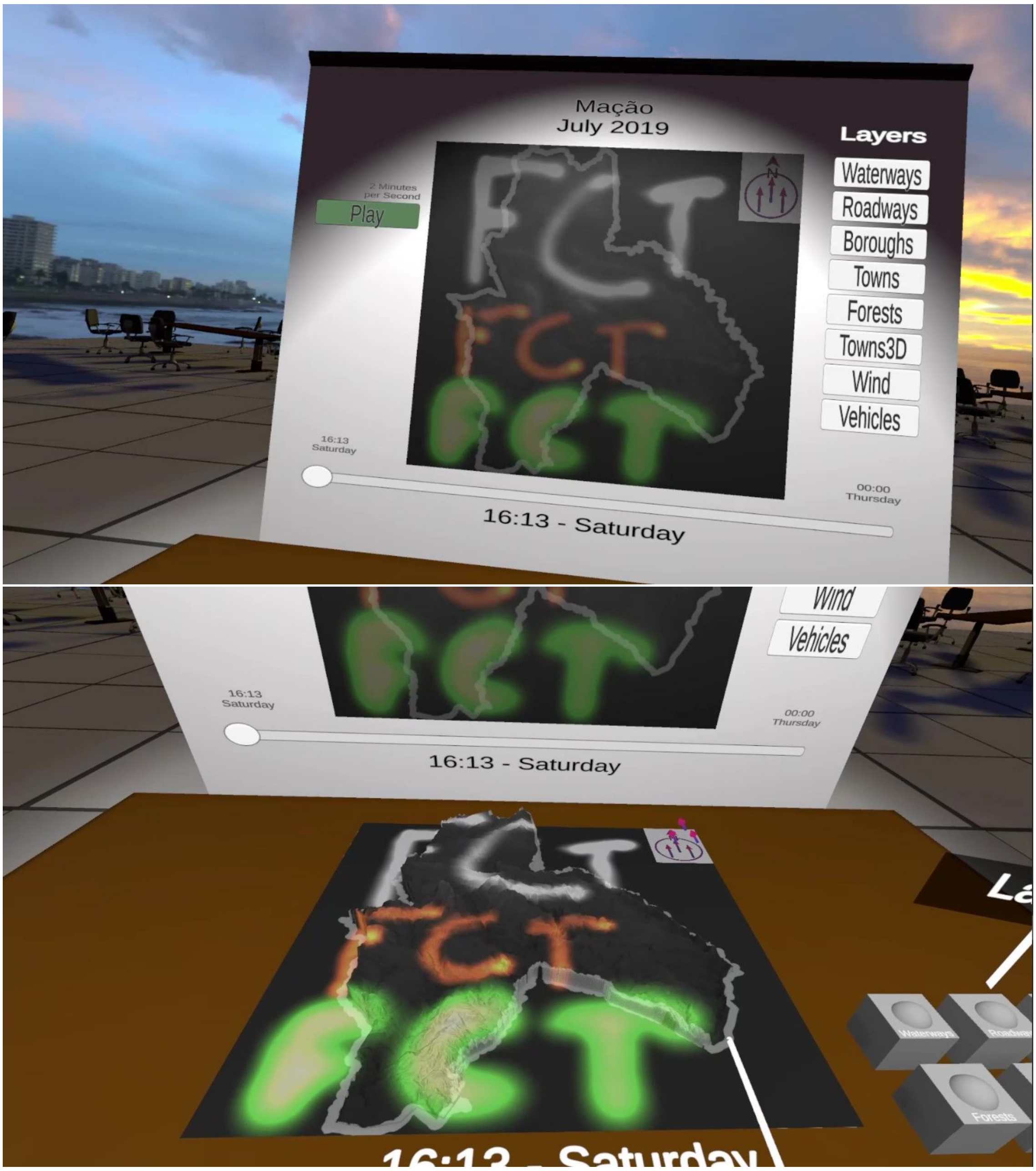


Figure 4.14: Brush costumizability in color, size and softness.

as fifteen minutes long then the vehicle will take fifteen minutes on the *Simulation Clock* to get to its destination.

User Component As with previous *User Components* this one is also only responsible for processing user's input and delegate the actual movement logic to the *System Component*.

System Component This is one of the most elaborate *System Components* implemented, alongside the *Ruler Tool*. It handles selection of vehicles, and making sure this selection is synced across clients that can collaboratively select multiple vehicles and see that reflected, as well as the logic for making a request to the OSRM server, parsing and turning its response into the necessary data structures to be visually represented and also traversable. Once the representation is known to the vehicle itself, it syncs with the *Simulation Clock* no longer needing intervention from this *System Component*.

4.3.4 Ruler Tool

In every situation a map is involved, at some point there arises the need to make measurements, either between two towns, a town and a vehicle, or any other combination of two points in the map. To account for that necessity the *Ruler Tool* was implemented. It is a flexible tool allowing to measure distances between two arbitrary points on the map. The user needs to point their controller to the place they want to measure, press the trigger then point to the second place and press the trigger again. The system will connect both points and add a label indicating the distance between them in kilometers. Each user has their own ruler, colored differently, and everyone sees everyone else's ruler, illustrated in Figure 4.15.

User Component As with previous *User Components* this tool is also only responsible for processing user's input and delegate the actual measuring logic to the *System Component*.

System Component The *System Component* is a bit more complex then the others. It is responsible for maintaining a correspondence between user and their ruler, which is composed by points and lines connecting those points which also must be tracked. Every time the user places one point to start a measurement, the *System Component* receives the *Command* from the *User Component* and must create a new measurement, associate it with the user, add this first point and send out a *ClientRpc* so all clients can do the same, ensuring synchronization of the user-ruler relationship and the ruler components.

4.3.5 Wind Tool

The *Wind Layer* was designed with the goal of visualizing how much risk a given change in the wind could affect the wildfire, as the fire can climb sharp inclines much faster if the wind helps. The *Wind Tool* supports the *Wind Layer* by allowing the user to control

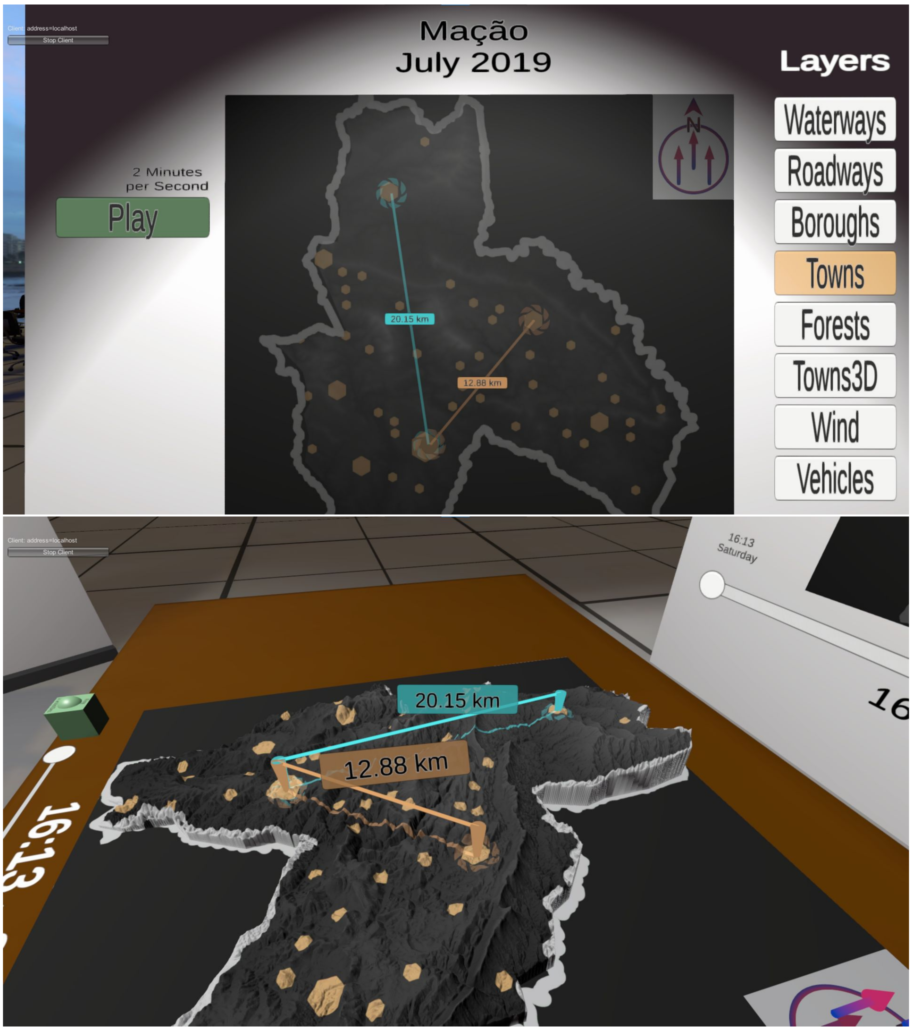


Figure 4.15: Two users measuring different distances simultaneously.

its visualization. This visualization depends on the wind direction provided by the user to color the terrain according to how exposed the terrain is to the given direction. The *Wind Tool* allows the user to determine this direction. How this visualization is achieved is detailed in Section 4.4, here only how the user interacts with the tool is described. Interaction-wise the *Wind Tool* is the simplest, in the sense that it does not do elaborate computations as the previous tools do and is just responsible for sending a direction over the network and updating it on the clients.

User Component This tool's *User Component* is solely responsible for sending user's input to the *System Component*. While using the *Wind Tool* the user can control the wind direction using the right controller's touchpad. The direction will be computed from the center of the touchpad towards the thumb's position. While the thumb rests on the touchpad this direction is being computed and sent over to the *System Component* using a *Command*.

System Component This tool's *System Component* is responsible for receiving the wind direction and updating the shader property accordingly, for all clients. Then the rendering takes over and the tool's job is done.

4.3.6 Simulation Clock

The *Simulation Clock* is a component that needs to be mentioned, as several elements of the VE (vehicle movement, marker lifetime and the fire playback itself) depend on it. Similarly to previous tools, the *Simulation Clock* has the two components. One that interprets and sends user's action on the clock and other that receives and updates every client's clock to be in sync. Every client has their own *Simulation Clock* that starts its lifetime in the paused state. The user can then *Play*, and *Pause*, the clock using the buttons available in the VE, upon button activation the *User Component* informs the *System Component* the user wants to start the *Simulation Clock* and it then starts every client's *Simulation Clock*. The user can also go forwards and backwards using the sliders also available in the environment which function similarly to the *Play/Pause* buttons. By relying on *System Component* to update every client's (local) clock the overall network traffic is reduced as the dependent systems can subscribe to these events locally and govern themselves by only taking into account the local *Simulation Clock*.

4.4 Rendering

This section details specifics on how the layers, fire and other elements were rendered in the VE. The application was developed using Unity's Universal Rendering Pipeline which means shader code would be written using Unity's Shader Graph visual editor. In the early stages this along with Unity's built in terrain shaders was enough, but when it came time to add more features to the 3D terrain, Shader Graph was a limiting factor. In order to bypass that, the base Unity terrain shaders had to be used and adapted to fit the project needs. At the end of the project there are, effectively, two different shaders that achieve the same visual result. One is the modified terrain shader applied to the 3D map and other is a 2D shader authored using Unity's Shader Graph. The techniques used are the same on both shaders and differ only on how they were authored (text-based versus node-based), thus no more distinction will be made.

With that out of the way, this section will elaborate on how the *geometric layers* are rendered, how the *Wind Layer* visualization was achieved, how the *Vehicle Layer* draws both a 3D representation and a 2D representation of the vehicles, followed by detailing how some tools' effects are represented, mainly the *Ruler* and *Brush* tools, finally the wildfire rendering will be explained.

4.4.1 Layer Rendering

Rendering the geometric layers is one of the simplest operations in the system. The complex steps to achieve it were executed during layer import using the custom tool *Layer Creator* mentioned in the previous chapter. Once the positions of whatever geometry is to be imported is calculated, the tool instantiates the preferred geometric primitives (small regular polygons for punctual geometry, polylines for line geometry and polygons for polygonal geometry), along with a camera with a *RenderTexture* attached and renders the scene to this texture and then saves the texture as an image file. This is a pre-processing step, once it is done, all that remains to do is to assign the texture to the terrain shader and sample it from there. Figure 4.16 shows the texture generated by the tool and the same texture projected onto the terrain.

4.4.2 Wind Layer Rendering

The wind layer was implemented with the intention to provide a visualization of the effects the wind would have on the terrain, and by extension the wildfire progression, as the wind factors quite heavily on it. For example, if a wildfire is climbing a slope and has wind on its back, it climbs much faster, thus making steep, wind-exposed slopes dangerous as they can aid the wildfire spreading.

With this in mind, an interactive visualization of how the wind affects the terrain was designed. The user can give a wind direction to visualize its effect on the terrain. In Figure 4.17, we can see the final wind visualization.

As referred in Section 4.1.4, the visualization needed access to terrain data regarding its *Slope* and *Aspect* that was then generated based on the terrain heightmap. Having

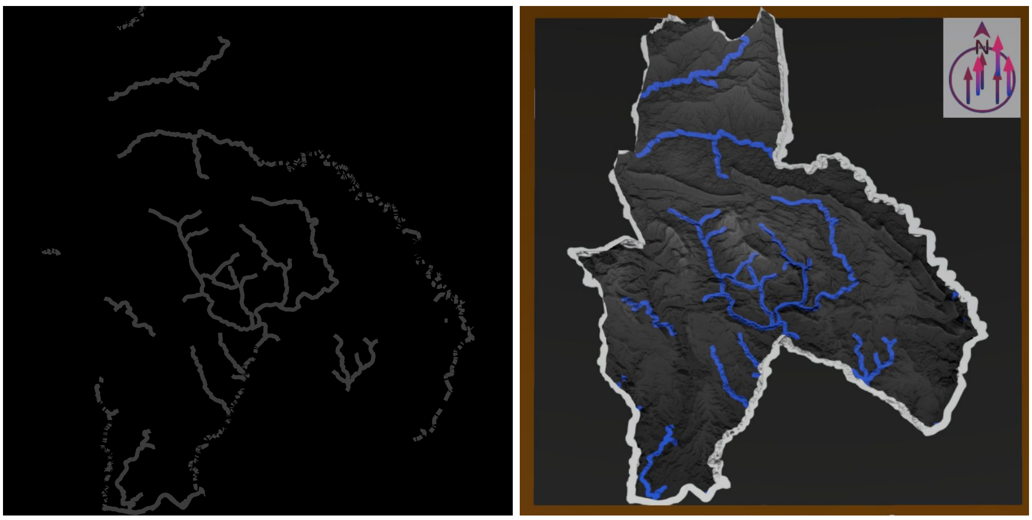


Figure 4.16: Left: The *waterways* texture generated by the *Layer Creator* tool. Right: Terrain rendering with the *Water Layer* turned on, showing that same texture being rendered.

this data, the color for each pixel of the visualization can then be calculated. One two-dimensional vector is constructed from the generated texture, representing the direction of the terrain, the *Aspect*. The *Slope* is also calculated from the generated textures. In order to avoid some small artifacts present in the generated textures, the value for each point is calculated as an average of their neighbors. One last important value is calculated the *Terrain Wind Exposure*, the dot product between the wind direction, provided by the user, and the terrain *Aspect* at that point. At this point, the color can be determined based on the previously calculated values. The *Terrain Wind Exposure* value is used to determine which regions get colored or not, meaning which regions are exposed to wind or not and the *Slope* value is used as the blend value for the green-red gradient. There are also some controls for this visualization that the user can try, such as a *Slope Threshold* that can be adjusted to only color slopes that are steeper than ten degrees, for example.

4.4.3 Tool Rendering

Some consideration should be given to rendering the visual elements that aid tool usage. As most can be used in both maps they should also be represented in both maps. Two tools will not be referred here, as one has just been detailed, the *Wind Tool*, and the *Marker*

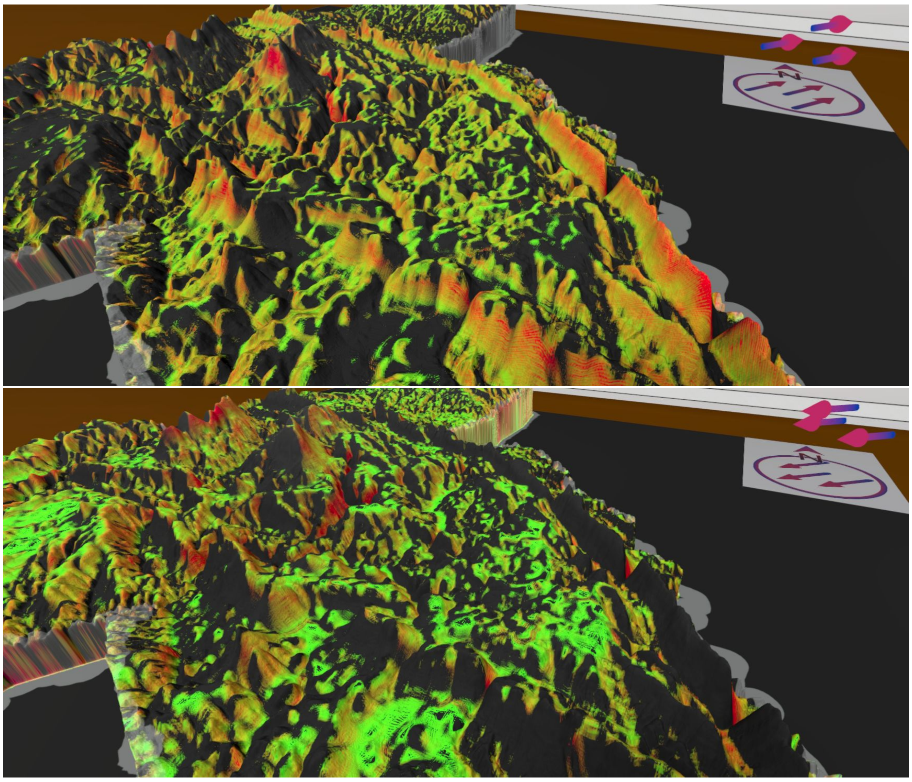


Figure 4.17: Wind visualization, wind direction indicated on top-right corner - Green regions represent plain terrain which has low exposure to current wind direction, while red regions represent slopes with high exposure to the current wind direction. Uncolored regions are sheltered from the wind.

Tool which can only be used in the three-dimensional map. Thus, this section will detail how the vehicles in the *Vehicle Tool* are represented, how the measurements from the *Ruler Tool* are communicated and how the scribbles from the *Brush Tool* are made and painted to the terrain. Some tools, as well as the *Wildfire Rendering* take advantage of a secondary camera, placed slightly above the three-dimensional terrain, oriented towards it, rendering to an off-screen texture that is then used on the terrain shader. This camera is very selective on what it renders, it only renders some necessary shapes, completely ignoring the three-dimensional terrain as it is already the most costly element to draw and would not be performant rendering it a second time.

Vehicle Tool The vehicles were first implemented solely as three-dimensional pin on the three-dimensional map and then changed to also have an indicator on the two-dimensional map. A sprite of the vehicle surrounded by a small indicator was added to the pin, facing the up direction. This sprite is not rendered to any of the main cameras and is instead rendered to the secondary camera previously mentioned. This allows it to be picked up by the two-dimensional map shader and rendered there at the correct position without any more book-keeping from the system. In Figure 4.18, the steps for this technique can be observed.

Ruler Tool The *Ruler Tool* functions similarly to the vehicle tool, by rendering some parts of its three-dimensional object to the secondary camera. One important difference is that this tool must communicate to the user the result of the measurement which would not work well rendering to the secondary camera. Instead, two labels with the result are rendered, one above the line's midpoint on the 3D map and another, also at the line's midpoint, on the 2D map.

Brush Tool This last tool has the most involved process for drawing its effects. It is responsible for filling one texture that is then read by the map shaders. Filling this one texture is the convoluted step the *Painting System* takes care of. An example of the process and the textures used for the effect can be visualized in Figure 4.19.

The main element of the *Painting System* is the *PaintableTexture* object. It stores three *Render Textures*, a *Command Buffer* and two references to some necessary *shaders*. The three *Render Textures* maintain information about current position in local terrain space, the accumulation of the brush strokes and the final texture that will be used in the terrain shaders to actually show the brush strokes. The textures are accompanied by simple shaders that are used when filling them, the first shader draws the place the user wants to actually draw as a circle affected by the brush parameters selected, then another shader is in charge of combining the brush position with the current drawing and then this accumulation texture is copied (*blitted*) over to the final texture to be used in the terrain shader.

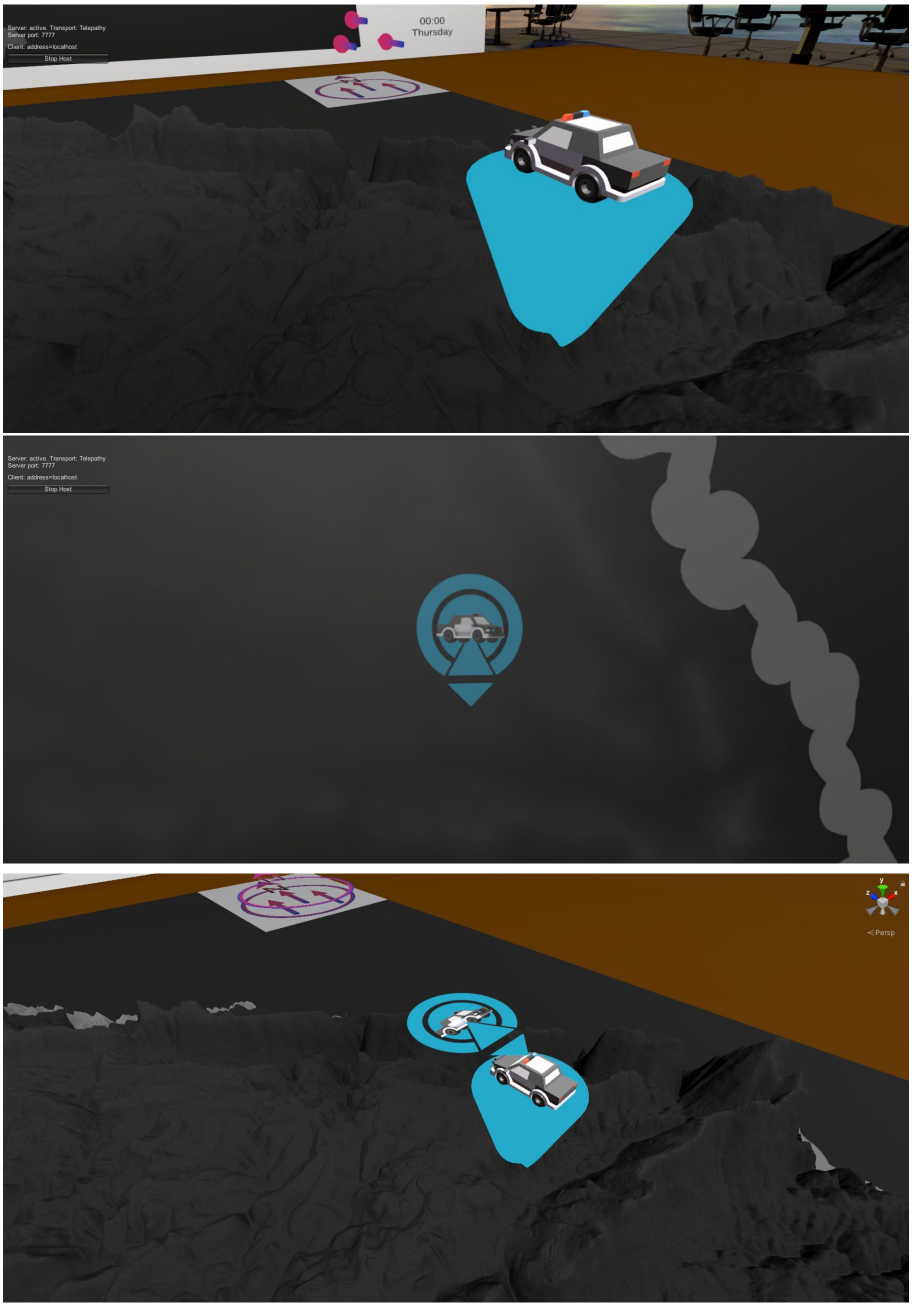


Figure 4.18: Three different close ups of the blue vehicle. Top: 3D Close up. Center: 2D Close up. Bottom: Close up of vehicle in editor, showing both the 3D version and the 2D version, the latter of which is only rendered to the secondary camera.

To avoid doing all this work every frame, the relevant draw calls were recorded into the *Command Buffer* allowing control over when they should be called, this means that filling these textures is only done when one of the users is actually using the *Brush Tool*, which is when the *Command Buffer* is executed.

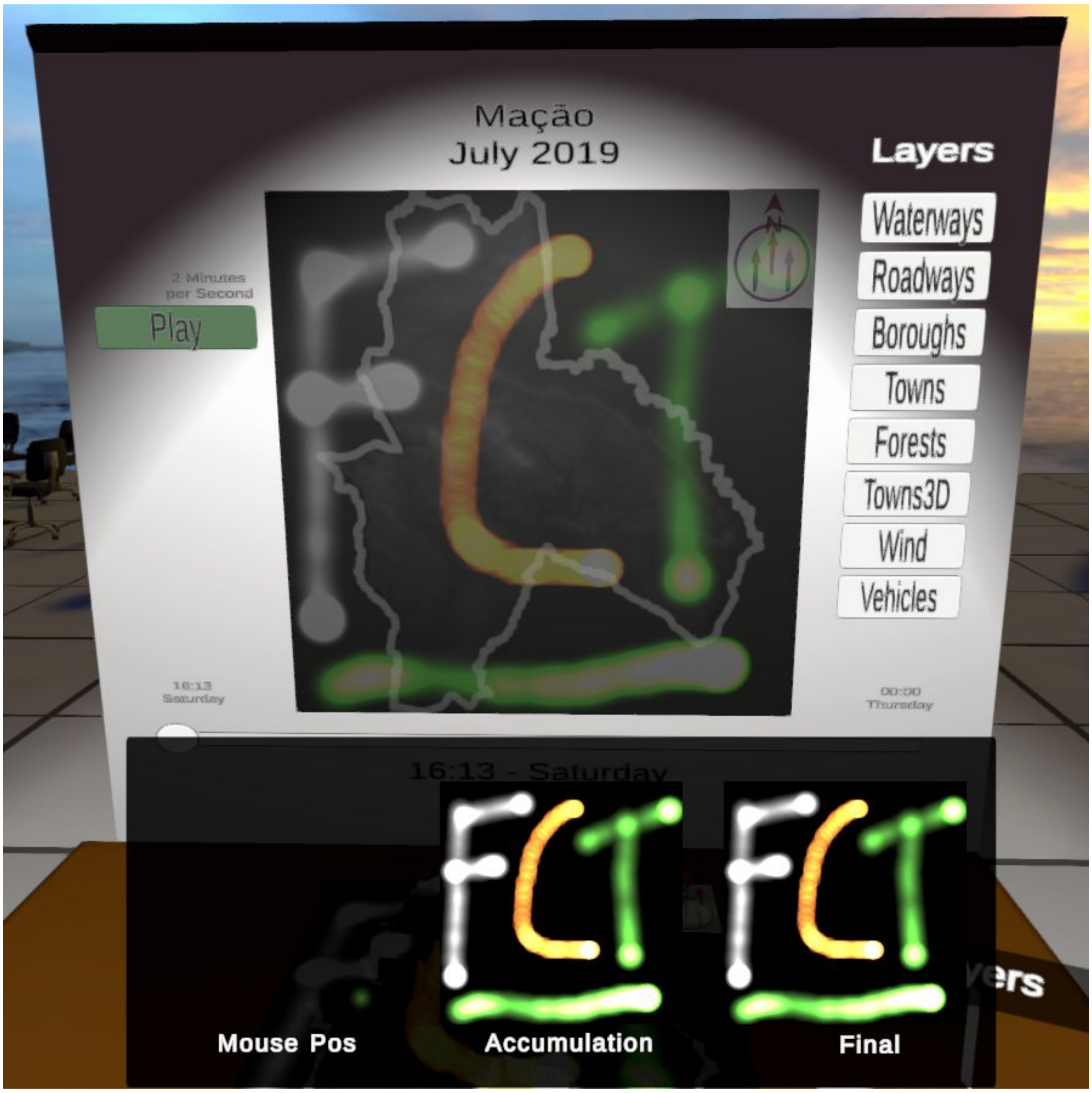


Figure 4.19: Initials of the University drawn onto the terrain. Below, from left to right are the buffers that feed the terrain shader. Special attention to the green point on *Mouse Pos* buffer, showing the last place the user painted (the line below the text, painted from left to right).

4.4.4 Wildfire Rendering

A necessary visualization for this particular use case is, of course, the wildfire itself. Actual simulation not being the goal of the project, instead the visualization relies on real data recorded during the wildfire. This data was not processed and is used as is, that means there are discontinuities in the fire fronts and burnt areas, fixing those is out of scope of this thesis.

The wildfire visualization has two components, the fire fronts and the burnt areas, represented by polylines and polygons respectively. These geometries are rendered to a secondary camera that then supplies its renderbuffer to the terrain shader so it can draw them directly on the terrain, in both 2D and 3D maps. This technique is similar to how the vehicles are represented in the 2D map, but applied to both maps. As the wildfire progresses, fire fronts may shift or extinguish and by subscribing to the *Simulation Clock*, fire fronts can turn themselves on or off which will make them stop drawing to the secondary camera and, consequently, to the terrain. Burnt areas behave similarly with the small distinction that these do not turn off after turning on. One more detail regarding the fire front rendering is that they serve as particle emitters, allowing a quick and visual distinction between fire front activity, in Figure 4.20 we can observe and visually distinguish high-activity, medium-activity and low-activity ones, with the high-activity emitting particles higher and having a brighter color as opposed to the close to the ground particles emitted by the low-activity front.

4.5 Collaborative Actions

Over the course of development lot of work went into making the VE as collaborative as possible. This section will detail how users collaborate within the application.

Users can connect to the same VE and be present in the same virtual space, they can see each other represented by an avatar and controllers for hands, as seen in Figure 4.21, this gives a sense of sharing the environment with another person and also aids in understanding where each user is looking at.

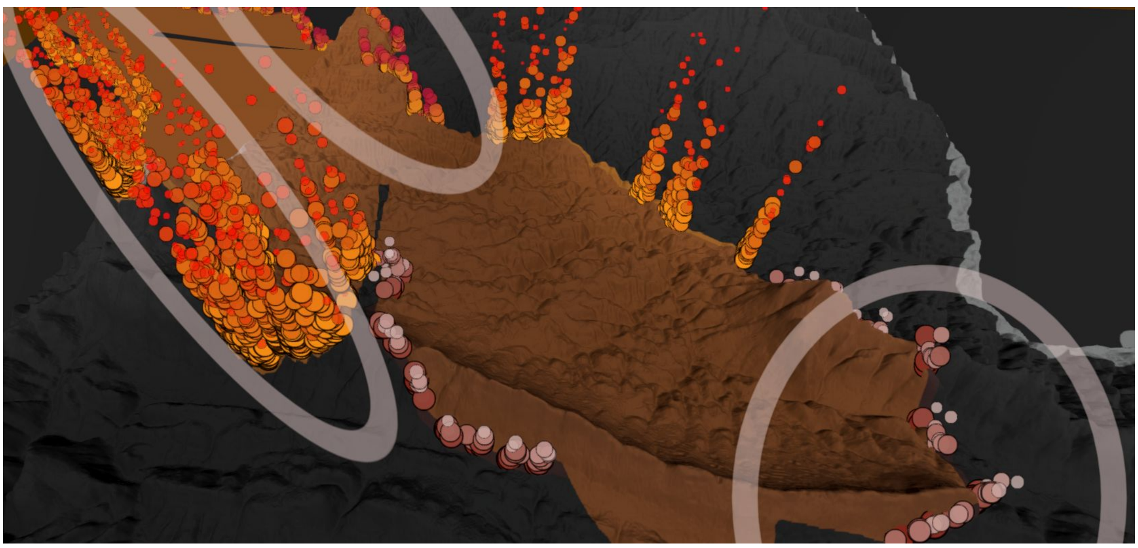


Figure 4.20: Close-up of wildfire visualization. Firefronts highlighted from left to right; High-intensity, medium-intensity, low-intensity.

The maps have several geolocated information associated with different layers. These layers are synced between clients.

The developed tools also take the shared space in mind, every tool action is reflected to all users. This aspect is represented slightly differently for each tool, these differences will be detailed below. Following this, the *Proposal and Notification* systems will be explained. This is a way for a user to highlight a given action; in the application the users can send notifications and approvals/denials for the markers placed on the 3D terrain and vehicle movement commands.

4.5.1 Tools

As mentioned before, tool actions are reflected to all users and behave slightly differently from each other.

Brush Tool Users can customize their brush's color, size and softness individually and this is reflected for both users in both maps, as illustrated earlier in Figure 4.14. The users can also clear the strokes, this clears the canvas for both users, as it is a shared canvas, akin to a classrooms whiteboard.

Ruler Tool Measurements taken with the *Ruler Tool* are represented for all users. Each user is assigned a different color and can take measurements independently from the other user, as seen in Figure 4.22. As opposed to the *Brush Tool*, once a user clears a measurement it only clears it for themselves, it does not interfere with other user's measurements.

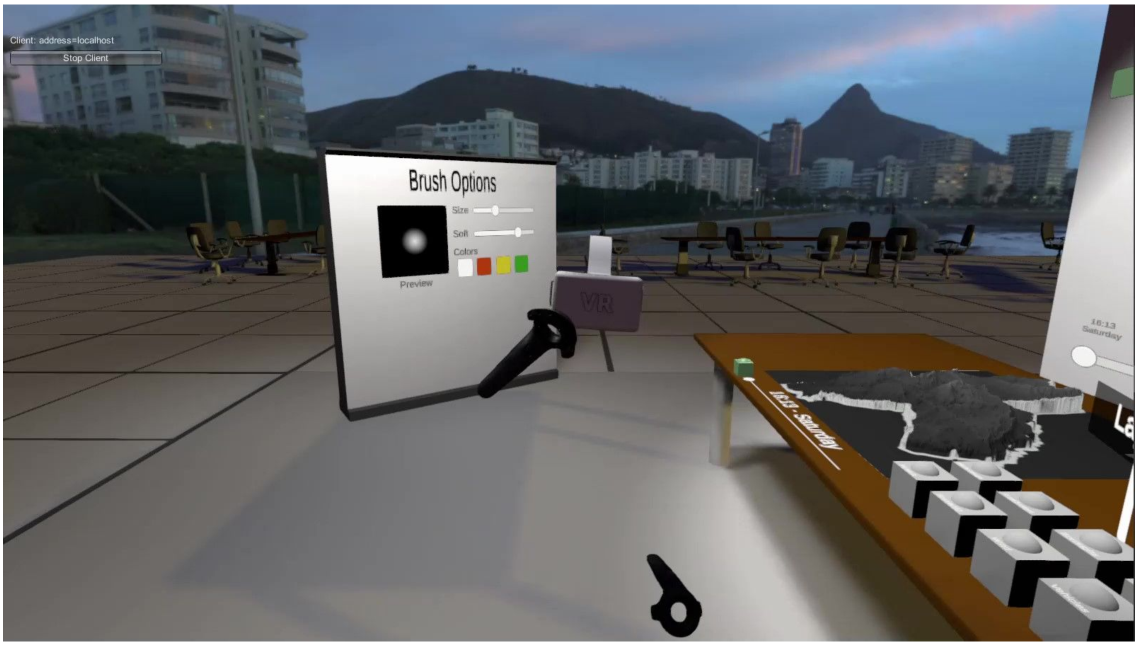


Figure 4.21: User waving to another in the same VE.

The *Wind Tool* is used to interact with the *Wind Layer* and does not have a collaborative behavior worth mentioning, besides the fact that once a user changes the wind direction it is reflected on all clients, as expected. Both *Marker Tool* and *Vehicle Tool* collaboration features will be detailed in the next subsection about the *Proposal and Notification* systems.

4.5.2 Proposal and Notification

The *Marker Tool* and *Vehicle Tool* have some interesting collaboration features, they take advantage of the *Proposal and Notification* systems. These two systems were designed to allow users to highlight actions to other users and ask for feedback on those actions. For example, a user intends to move a vehicle to a particular place - first, using the vehicle tool the user proposes a destination to the other user, second, the other user receives a notification on the controller, is somehow made aware of where the proposed destination for the vehicle lies and then, finally, is prompted to accept or deny this proposal. For the second step three different techniques of "making the user aware of the proposed destination" were implemented and tested.

The three techniques are distinct in how they point the user to the relevant place, from now on referred to as *the marker*; one is controller centric, other is destination centric and the remaining one is both, connecting controller and destination. They are: *Arrow* which is controller centric; *Line* which connects both controller and destination; *Sphere* which is destination centric.

Arrow The *Arrow* notification is controller centric, meaning its indicator is fixed on the users right controller. This implies the user must look at the controller, understand

Figure 4.22: Both users measurements are represented on both maps for both users.

where the arrow is pointing to and then look at the *the marker*. Figure 4.23, illustrates this notification.

Line The *Line* notification connects both controller and destination with a line, improving on previous method by saving the user the mental task of guessing to which marker the *Arrow* is pointing, this might be simple with a low amount of markers in place but gets challenging as the number of markers present grows. The line directly connecting to *the marker* makes the new marker immediately apparent. Figure 4.24 illustrates this technique.

Sphere The *Sphere* notification is destination centric, meaning a disconnect between the user's controller and the highlighted *marker*, this might or might not be an issue. It is represented as a transparent sphere englobing the user and shrinking down to *the marker's* position, prompting the user to follow the shrinking movement. An example of this can be seen in Figure 4.25.

Figure 4.23: Close up of the Arrow notification.

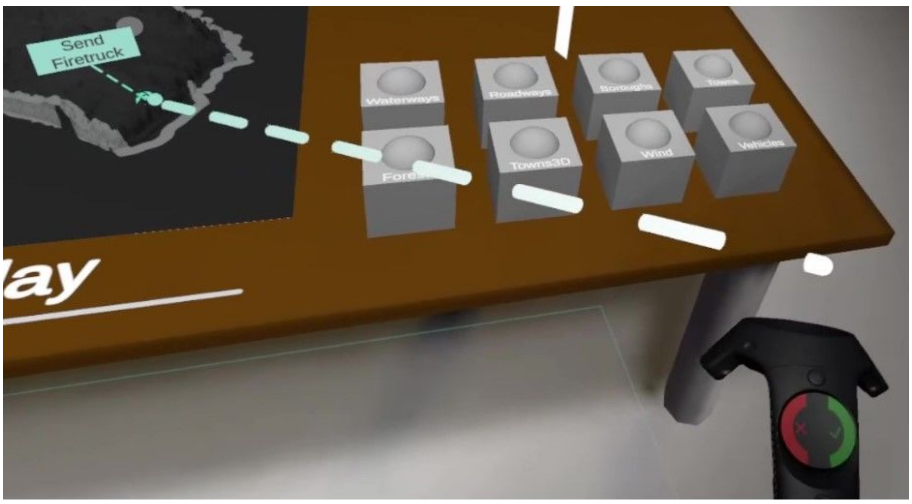


Figure 4.24: Close up of the Line notification and the indication for approving or denying the proposed marker.

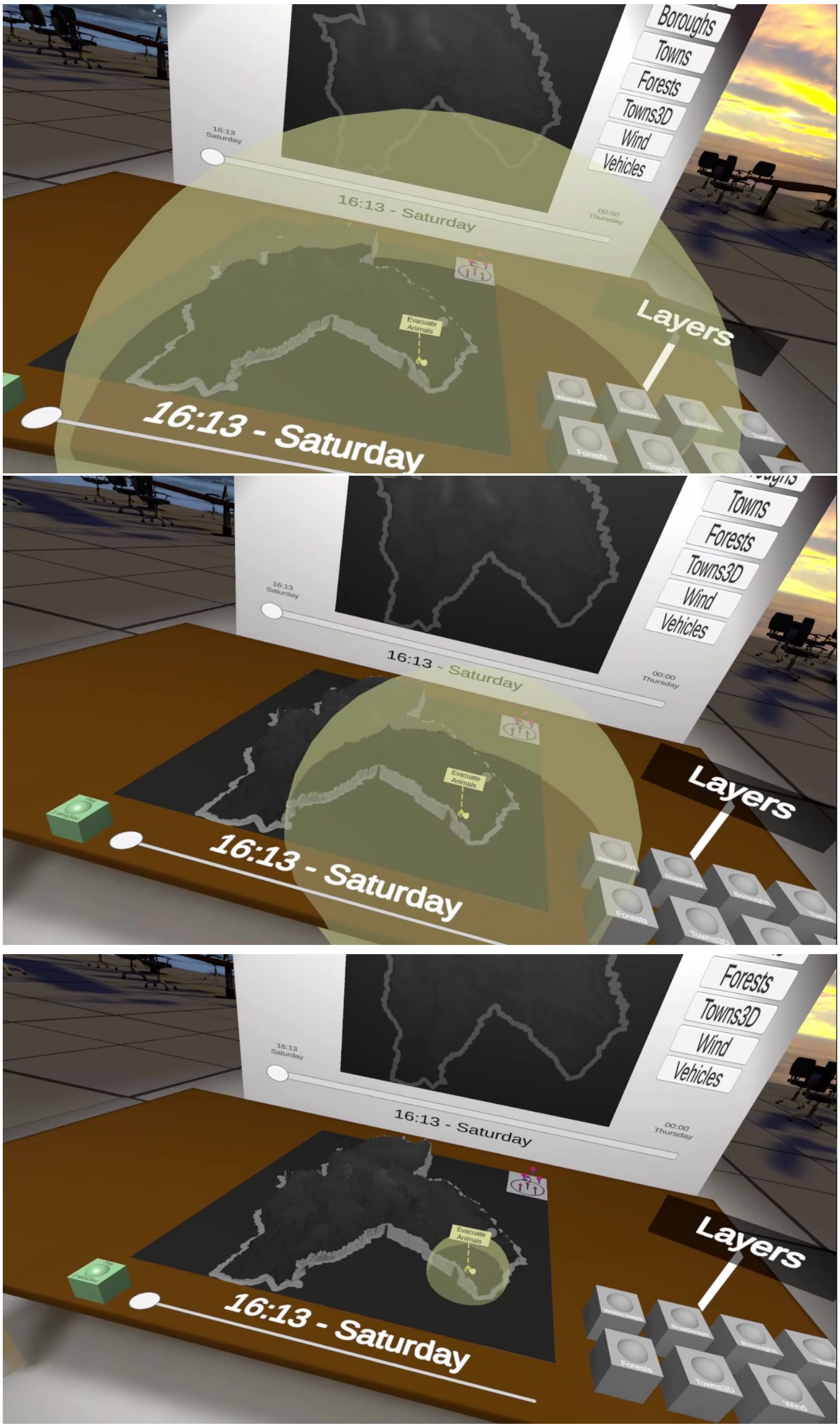


Figure 4.25: Illustration of the Sphere notification shrinking centered on the marker.

These three techniques were tested and the results are discussed in Chapter 5.

The last element of the *Proposal and Notification* systems, is the acceptance of any proposal. If a user makes several proposals they are presented to the other user in a *First In First Out* fashion and each one must be accepted or refused before the next one is shown. Each "shown" proposal always start by notifying the user using one of the techniques already mentioned. Once the user is ready to accept or deny the proposal they must simply press the directional button corresponding to the desired action, as shown earlier in Figure 4.24.

4.6 Summary

This chapter detailed the development of this thesis' application. The chapter went into detail about the *Georeferenced Data* that was necessary to get all information needed into the system and how each piece of data is characterized; heightmap terrain data, geometry and supporting data used for layers, the OSRM and the wildfire data itself. Then the *Networking* components were explained followed by how they were used for implementing the *VR Interface*. After that, a brief explanation on how the wind visualization was implemented and how other map elements and tool effects were rendered. *Collaborative Actions* closed the chapter, where details on how specific tools function in a collaborative setting followed by a description of the *Proposal and Notification* systems.

The final product of this implementation is a multi-user collaborative VR system, with a dual-map setup to display georeferenced data. It features:

- VR Interface:
  - Dual-map setup;
  - Marker Tool;
  - Brush Tool;
  - Vehicle Tool;
  - Ruler Tool;
  - Wind Tool.
- Data Visualization:
  - Georeferenced layers;
  - Wind impact visualization;
  - Temporal wildfire visualization.
- Collaboration:
  - Multiple users connected to the same VE;

- Proposal and Notification System;
- Every action is shared and visualized by all.

Chapter 5

Evaluation

To answer the proposed research questions in Section 1.2 and validate the system developed over the course of this thesis user tests were conducted. This chapter explains the protocol followed conducting the user tests, what kind of data was recorded and what insights it revealed. Starting by detailing the testing protocol and how each task is tested and what questions are made, then once that is explained, the data analysis conducted is described along with a discussion of the results for each task. By the end, a section is dedicated to a more general discussion and analysis of the results.

It was decided to test the multiple elements of the interface, focusing on wether the users were able to interact and use all the components available, what tools performed better in which map, which type of notification performed better with the users, as well as outside impact (previous experience with Virtual Reality (VR)/videogames) on performance tests. Most of this work was developed during the Covid-19 pandemic, thus the collaborative tests were done with the researcher due to restrictions in place.

5.1 Protocol

The test session required two people to be physically present in the same room, the researcher and the test subject. The test subject starts by reading the informed consent, present in Appendix B, and agreeing to it. Right after, the subject is given a small tutorial on how to use the controllers, with emphasis on having one hand dedicated only to toolbox management and the other to tool usage and general interaction. Then the user puts on the already disinfected VR headset and controllers and gets into the Virtual Environment (VE). The researcher is also connected to the same VE using a keyboard and mouse and gives the subject a quick virtual tour of the environment. The subject then gets a several minutes to explore the environment and then the tasks start.

The tasks are small actions the user is asked to make, such as activating a layer or taking a measurement. These are grouped into larger blocks of tasks, here referred to as groups, done sequentially, and each group tests a related system. For each task, the researcher fills a form with small observations and, once the task is completed, he asks the subject some questions related to completing said task. These questions and observations are detailed below for each task. Most data is collected manually by the researcher apart from the *Notification System* group tasks, where the application itself collects the relevant data. For some questions, explicit comparisons are made between either the two available interfaces, *Wall* and *Table*, or between the three notification types, *Arrow*, *Line*, and *Sphere*. T-tests were used to compare differences within interfaces and ANOVA test to compare within notification types.

The research questions to which this thesis intends to answer are stated here, once more. Overall, the tasks participants executed intend to help answer all the questions. Particular task groups are more pointed towards particular questions. Task group *B* and *D* and *E* test the dual-map interface, searching for an answer specifically to question *Q1*. Task group *C* tries to find an answer to *Q3*. Finally, all task groups contribute to insights related to the main question, *Q*, while generating data helpful in to better understanding answers both *Q2* and *Q4*.

- Q - Can a crisis management scenario be improved in a collaborative VR environment?
- Q1 - Are there benefits to a dual-map (2D/3D) visualization?
- Q2 - Is there a preference for one interface over the other?
- Q3 - How can users notify each other of new information in the system?
- Q4 - Are users with no VR experience able to adapt quickly to the VE and the available tools?

5.2 Group A - General Exploration

The first group of tasks focuses on helping the user getting familiar with the environment and how they will interact with it. It teaches the user about the concepts necessary for later tests, such as activating layers, selecting and using the available tools and how to locate towns in the map.

5.2.1 Task A1 - Activate "Waterways" and "Roadways" Layers

The first task the user is instructed to complete is to activate two layers. No more direction is given and the user themselves chooses to either use the wall interface or the table interface. This task has no questions for the subject, but some observations are written down by the researcher.

A1Q1 - Able to complete? Answer can be *Yes*, *Yes, with help* or *No*.

A1Q2 - Time to completion Answer is time to completion in seconds.

A1Q3 - Interface used Answer is either *Wall* or *Table*.

A1Q4 - Observations The researcher can write here some relevant observations made by either person during the study.

These questions are meant to understand how easy it is for a new user to execute the most basic task in the VE. It also gives some insight into the tendency of using the *Wall* buttons or the more tactile *Table* buttons.

5.2.2 Task A2 - Activate "Wind" Layer and Use the "Wind" Tool

For this task the user must activate and interact with the *Wind Layer* using the *Wind Tool*. Once again, the user chooses which map to observe. Some questions regarding understating of the colorization of the terrain are asked.

A2Q1 - Able to complete? Answer can be *Yes*, *Yes, with help* or *No*.

A2Q2 - Time to completion Answer is time to completion in seconds.

A2Q3 - Interface used Answer is either *Wall* or *Table*.

A2Q4 - What does the colorized terrain mean? Open question where the user must analyse the visualization, understand what it means and how they are controlling it.

A2Q5 - What do the red colored areas mean? Multiple choice questions where the user can pick between *Terrain more exposed to the wind direction*, *Terrain less exposed to the wind direction* and *Terrain more exposed to Northern wind*.

A2Q6 - Observations The researcher can write here some relevant observations made by either person during the study.

These questions evaluate how well the wind visualization communicates its meaning to the user. It also serves as an introduction to the tool selection wheel and how to use a tool to interact within the VE.

5.2.3 Task A3 - Locate "Amêndoa" and "Cardigos" on the Upper Half of the Map

This task teaches the user how they can point to towns in the map and see what they are named. These particular towns are then used on later tests, so this also serves to teach the user where these towns are so they don't have to look for them again.

A3Q1 - Able to complete? Answer can be *Yes*, *Yes, with help* or *No*.

A3Q2 - Time to completion Answer is time to completion in seconds.

A3Q3 - Interface used Answer is either *Wall* or *Table*.

A3Q4 - Observations The researcher can write here some relevant observations made by either person during the study.

This task also relies on the user understanding that they must activate the relevant layers to discover where the towns are located.

5.2.4 Task A4 - Place a Marker Ordering Evacuation of Civilians near "Amêndoa"

To complete this task the user is asked to place a specific marker near the town "Amêndoa", located in the previous task. This task must be completed using the table interface.

A4Q1 - Able to complete? Answer can be *Yes*, *Yes, with help* or *No*.

A4Q2 - Time to completion Answer is time to completion in seconds.

A4Q3 - Observations The researcher can write here some relevant observations made by either person during the study.

Besides reinforcing how to use the tool wheel and interacting by pointing and clicking, the user is also presented with the marker concept, used later in the *Notification System* tests. It is important for the user to know what a marker is and what to look for once that test arrives.

5.2.5 Task A5 - Move Wildfire Playback to Monday

This task asks the user to progress the wildfire to some days later, making it clear that the fire is spreading rapidly and that the markers are tied to the specific time (and place) they were placed.

A5Q1 - Able to complete? Answer can be *Yes*, *Yes, with help* or *No*.

A5Q2 - Time to completion Answer is time to completion in seconds.

A5Q3 - Observations The researcher can write here some relevant observations made by either person during the study.

This task concludes the first group, with the user having a general understating how the various elements of the system work together. They haven't tried every tool but by now should have an idea how to interact with the remaining tools.

5.3 Group B - Layer Activation

At this point, with the user already comfortable using the system, the tasks shift focus. This particular group intends to test the *Wall* interface and the *Table* interface in order to understand which the users prefer and which is more efficient. Since the questions are similar throughout the group and the tasks themselves are similar, they are all described in this section. The tasks first.

Task B1 - Activate “Vehicles” Layer using the Wall Interface

Task B2 - Activate “Forests” Layer using the Wall Interface

Task B3 - Deactivate “Vehicles” Layer using the Table Interface

Task B4 - Deactivate “Forests” Layer using the Table Interface

And the questions after.

BQ1 - Time to completion Answer is time to completion in seconds.

BQ2 - “I considered it comfortable” The user classifies this affirmation with a one to five number, with one meaning *Strongly Disagree* and five meaning *Strongly Agree*.

BQ3 - Observations The researcher can write here some relevant observations made by either person during the study.

BQ5 - Interface Preference The last question of the group asks the user which was the preferred interface to activate/deactivate the layers.

These questions are meant to evaluate which of the two interfaces the user prefers to use and which are more efficient to use, which the answer might not be the same to both. The tasks are similar varying only on which interface the user should use and which layer should they activate/deactivate. Consequently, the questions and observations are the same in every task. To avoid repetition, it should be taken into account that every question, *BQ1*, *BQ2* and *BQ3* were asked/recorded for every task, *B1*, *B2*, *B3* and *B4*, while *BQ3* was asked after all tasks were completed.

5.4 Group C - Notification System

This group is focused on testing how to highlight an element of the VE to another user. These tasks use the *Notification System* described in Chapter 4.5.2. The three tasks are always the same and so are the questions. What varies is how the notification is presented to the user. The notification can be an *Arrow*, a *Line* or a *Sphere*. The order in which they appear to the test subjects follows the Latin Square Order, which means it is not always the same for each user. The user must also, accept or refuse a marker, by pressing the corresponding direction on the controller touchpad, in order to complete each task.

In order to test the three different notification types, the users are asked to rotate 90 degrees clockwise and take one step forward, so as they are facing away from the terrain. Then, they are informed that the researcher will place a marker somewhere on the map which will cause them to receive a notification and once they perceive this notification they should locate the marker itself and then accept or refuse it.

Task C1 - Notification by Arrow

Task C2 - Notification by Line

Task C3 - Notification by Sphere

Similarly to the previous group, the questions are asked/recorded for every task; *C1*, *C2* and *C3*. The last question, *CQ7*, refers to the whole group and not an individual task.

CQ1 - Did the user understand the notification? *Yes* or *No* question answered based on researcher observation.

CQ2 - Did the user find the marker? *Yes* or *No* question answered based on researcher observation.

CQ3 - Time to find the marker Time it took for the user to find the marker and accept/decline the marker, automatically collected by the system.

CQ4 - "I found the marker easily." The user classifies this affirmation with a one to five number, with one meaning *Strongly Disagree* and five meaning *Strongly Agree*.

CQ5 - "The notification is visible." The user classifies this affirmation with a one to five number, with one meaning *Strongly Disagree* and five meaning *Strongly Agree*.

CQ6 - Observations The researcher can write here some relevant observations made by either person during the study.

CQ7 - Notification Preference The last question of the group asks the user to order the notification types according to personal preference.

The *Time* it took the user to complete, the user *Position*, *Rotation* and *Forward Vector* of the head, the *Right Controller Position* and the *Marker Position* are all auto-collected data the system saves after each task.

5.5 Group D - Ruler Tool

The tasks belonging to *Group D* are focused on testing the *Ruler Tool*. The user is asked to use the tool to measure distances between two points-of-interest, a town and a vehicle. The users will be instructed to either use the *Table* map or the *Wall* map and each testing session will start by using a different map. Similarly to the previous task group, the questions are almost equal and will not have their own subsection. They also follow the scheme already established, with all questions being asked/recorded for all tasks.

Task D1 - Measure the Distance Between "Cardigos" and the Red Vehicle Using the Wall Map

Task D2 - Measure the Distance Between "Amêndoa" and the Blue Vehicle Using the Wall Map

Task D3 - Measure the Distance Between "Cardigos" and the Red Vehicle Using the Table Map

Task D4 - Measure the Distance Between "Amêndoa" and the Blue Vehicle Using the Table Map

DQ1 - Time to completion Answer is time to completion in seconds.

DQ2 - Measurement value Value measured by user in Km.

DQ3 - "I considered it comfortable" The user classifies this affirmation with a one to five number, with one meaning *Strongly Disagree* and five meaning *Strongly Agree*.

DQ4 - Observations The researcher can write here some relevant observations made by either person during the study.

DQ5 - Map Preference At the end, the user is asked which of the two maps is preferred.

This task group aims to evaluate the *Ruler Tool* and how it behaves on both 2D and 3D maps. The act of measuring is the same on both maps, pointing and clicking, but having a 3D map can bring some difficulties. The value measured is also compared to the true value, defined by a precise measurement taken by the researcher using the keyboard and mouse client, to evaluate how accurate this particular implementation of the *Ruler Tool* is.

5.6 Group E - Brush Tool

This task group is meant to evaluate the *Brush Tool* and the painting system. The user is first asked to identify the town highlighted by the researcher, through the use of the brush, and then asked to highlight a town of their choosing.

5.6.1 Task E1 - Identify the Highlighted Town

The researcher highlights the capital, Mação. The user then must locate the highlighted town.

E1Q1 - Was able to identify the town? *Yes* or *No* answer.

E1Q2 - Time taken to identify the town Time the user took to identify the town, measured by the researcher.

E1Q3 - Observations The researcher can write here some relevant observations made by either person during the study.

5.6.2 Task E2 - Highlight a Town

The user is asked to highlight a town of their own choosing using the interface, *Wall* or *Table*.

E2Q1 - Was able to highlight a town? *Yes* or *No* answer.

E2Q2 - Time taken to highlight the town Time the user took to highlight the town, measured by the researcher.

E2Q3 - Interface used Answer is either *Wall* or *Table*.

E2Q4 - Observations The researcher can write here some relevant observations made by either person during the study.

5.7 Group F - Supervisor and Subordinate Scenario

Group F is the last group and is focused on a collaboration scenario. It is composed of a series of steps rather than discrete tasks. The two participants, the researcher and the test subject are playing the roles of *Supervisor* and *Subordinate*, respectively. The *Supervisor* issues an order by placing a marker on the terrain. The *Subordinate* must accept said order by locating the marker and accepting it. Then, after reading the marker that says "Send Firetruck", the *Subordinate* must select the correct vehicle and propose sending it to the desired location. Finally, the *Supervisor* must accept the *Subordinate*'s proposal and then the scenario ends.

FQ1 - Scenario completed? *Yes* or *No* answer.

FQ2 - Time to complete the scenario Time the scenario took to complete, from the initial marker placement by the researcher to the user proposing the route for the vehicle.

FQ3 - "I understood what was asked." Answer is a number from one to five, meaning *Strongly Disagree* and *Strongly Agree*, respectively.

FQ4 - "I understood I was sharing the VE with another person." Answer is a number from one to five, meaning *Strongly Disagree* and *Strongly Agree*, respectively.

FQ5 - Observations The researcher can write here some relevant observations made by either person during the study.

This task starts with the researcher explaining the scenario and then informally asking the user how they would select the vehicle. Since the selection of the vehicle is different than every other interaction up to this point, the user is instructed to select the vehicle by touching it and the proposing a route by pointing and clicking. As an aside, the majority of users were confused by this and mentioned their initial tendency to point and click for selection and then point and click again to propose a route.

This task concludes the practical component of the testing session. At this point, the user takes of the VR headset and is asked to complete the questionnaires related to *Usability*, *Presence*, the *System* itself and some *Characterization* questions.

5.8 User Questionnaires

After completing the tasks the users were ask to complete some questionnaires, evaluating different facets of the system and giving some more information regarding participant characterization. The questionnaires are described in the next sections and their questions are presented in annex when appropriate.

5.8.1 System Usability Scale (SUS) Questionnaire

The first questionnaire the user answers is the SUS Likert-scale questionnaire. Developed by John Brooke [5], it consists of ten questions the user must answer by assigning to each a value between one and five, meaning *Strongly Disagree* and *Strongly Agree*, respectively. The questionnaire yields a score between 1 and 100 points but shouldn't be interpreted as a percentage mark, but can be used to do a rough comparison between interfaces. According to the author on a later publication, John Brook [6], mentioned Jeff Sauro's research[1] where they mentioned that a score of 68 is the average score of the over 3.500 SUS results investigated.

The questionnaire is replicated in Annex I.

5.8.2 Igroup Presence Questionnaire (IPQ)

The second questionnaire is the IPQ. Developed by Thomas Schubert [37], it evaluates the "sense of being there" in a virtual environment by measuring three axis, *Sense of Spatial Presence*, *Involvement* and *Experienced Realism*. The questionnaire is composed of fourteen questions which, as before, the user must answer using a Likert-scale.

The questionnaire is replicated in Annex II.

5.8.3 System Questionnaire

The *System Questionnaire* has only two questions and serves to understand how the user felt using the shared VE, if they understood they were sharing the virtual space and if

[1] Measuring Usability With The System Usability Scale (SUS) Jeff Sauro's article on interpreting SUS Scores. - Last accessed 04/05/2021

they think they could use it as a collaboration tool with other users.

The questions are answered with a Likert-scale, using a value between one and five, where one meaning *Strongly Disagree* and *Strongly Agree*, respectively.

SQ1 - I was aware I was sharing the VE with another person.

SQ2 - I consider I could use this system to collaborate with another person.

5.8.4 Characterization Questionnaire

The last questionnaire the user is asked to fill is about themselves. It asks relevant details about the participant in order to be able to characterize the population used to test the system.

The questionnaire asked about participants *age*, *gender*, *height*, *completed education*, *dominant hand*, *VR experience*, *videogame experience* and which kind of game and platform, *experience with Emergency Management Systems* and wether the participant *had difficulty seeing*.

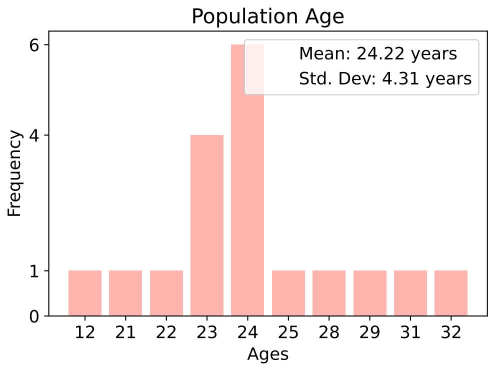
A more detailed description of these questions can be found in Appendix A.

5.9 Results and Analysis

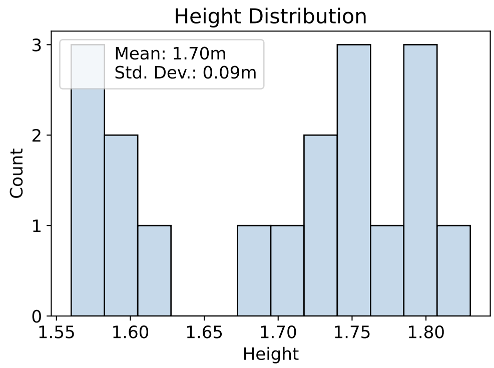
This section will go over the results and insights obtained. Starting with *Population Characterization* and then following the order the tasks were performed. Each task will first be analysed on its own, in an independent context and by the end of the section relevant tasks will be analysed taking account more factors, such as other tasks and population characteristics. After analyzing the tasks, the results of both the IPQ and the SUS questionnaires will be interpreted. Concluding the section, there will be an overall discussion relating interesting data points.

5.9.1 Population Characteristics

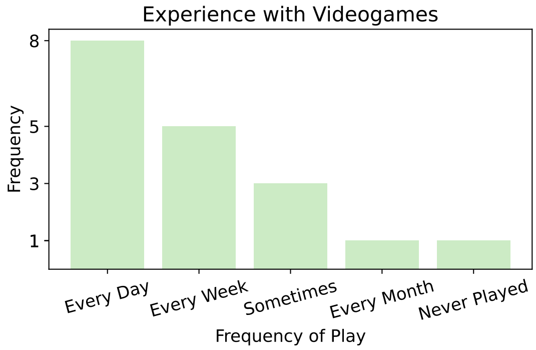
The population was composed of 18 participants, their demographics are present in Figures 5.1 and Tables 5.1. Participants gender are present in Table 5.1a and their ages vary between 12 and 32, with a median of 24, Graph 5.1a, with a height distribution present in 5.1b. Most participants were right-handed (95%), with only one being left-handed (5%), Table 5.1b. The education level varies, but 88% has completed some level of higher education, Table 5.1c. Most participants (89%) had no experience with emergency management systems, with only 2 having some (11%), Table 5.1d. All participants, except one, had previous experience with videogames, present in Figure 5.1c and the games played present in Figure 5.1d, while 50% had previous experiences with VR. No participants reported difficulty seeing even when using glasses (due to wrong prescription, older glasses. . .).



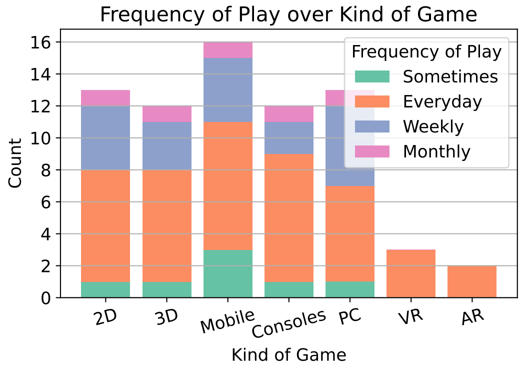
(a) Population Age



(b) Population Height



(c) Experience with Videogames



(d) Frequency of Play

Figure 5.1: Population demographics graphs.

5.9.2 Group A: Task A1 - Activate "Waterways" and "Roadways" Layers

Group A is an introduction to the system and intends to get the participant familiarized with it. The tasks are based on simple interactions with the environment and use of tools. An interesting data point that will be observed over the course of these tasks, and analysed later, is participant's natural tendency to use on interface over the other.

Table 5.1: Population demographics tables.

(a) Participants Gender

Gender	Participants (#)
Male	10
Female	8

(b) Participants Dominant Hand

Dominant Hand	Participants (#)
Right-Handed	17
Left-Handed	1

(c) Participants Education Level

Education Level	Participants (#)
Primary	1
Secondary	1
Bachelor's	12
Master's	4

(d) Participants Experience with Emergency Systems

Has Experience	Participants (#)
Yes	2
No	16

The first task introduces the participant to the system and the concept of layers and how to turn them on and off. The participant can use either the *Table* interface or the *Wall* interface. Data from this task is synthesized in Table 5.2. Participants were observed wether they could complete the task alone, with help or not as seen in Table 5.2a. The time to complete the task was measured and is presented in Table 5.2b, since participants were not instructed to choose a specific interface to interact with, but were aware they could use both, the choice of interface was also recorded, in Table 5.2c.

Table 5.2: Task A1 - Results

(a) A1Q1 - Able to Complete

Completed	Answer (#)
Yes	17
Yes, with help	1
No	0

(b) A1Q2 - Time to Completion (seconds)

Interface Used	$\bar{x} \pm \sigma$ (s)
Table	4.66 ± 0.62
Wall	4.99 ± 3.38
Combined	4.92 ± 2.97

(c) A1Q3 - Interface Used

Interface	Frequency (#)
Table	4
Wall	14
Combined	18

A two-sample t-test was used to determine if the use of interface, *Wall* or *Table*, had a significant impact in the participants' completion time. Taking into account that participants were free to choose which interface to use, the samples for each were not in equal size. Because of this, a two-sample Welch t-test was used as it is more reliable for uneven sample sizes [34]. The time to complete the task using the *Table* interface was lower (4.99 ± 3.38 seconds) than the time to completion using the *Wall* interface (4.66 ± 3.38 seconds). There is no evidence ($t(15) = 0.35, p = 0.73 > 0.05$) supporting the use of one interface over the other for speed of use, considering a confidence interval of 95%. One interesting observation, that will be discussed later, is the participants' natural tendency to pick one interface instead of another and how that tendency changes over the course of the tasks.

5.9.3 Group A: Task A2 - Activate "Wind" Layer and Use the "Wind" Tool

This second task introduces the concept of tools to the participant. Data regarding task A2 is shown in Table 5.3. The participants were observed on wether they could complete the task, Table 5.3a. How long it took them to complete it, Table 5.3b, and which interface they focused on, Table 5.3c. Then two question were asked; one open question and one multiple choice question, analysed below.

A two-sample t-test was used to determine if the use of interface, *Wall* or *Table*, had a significant impact in participants' completion time. As before, the participants are free to

Table 5.3: Task A2 - Results

(a) A2Q1 - Able to Complete

Completed	**Answer**
Yes	15
Yes, with help	3
No	0

(b) A2Q2 - Time to Completion (seconds)

Interface Used	$\bar{x} \pm \sigma$ **(s)**
Table	20.84 ± 06.46
Wall	28.83 ± 16.44
Combined	25.72 ± 13.77

(c) A2Q3 - Interface Used

Interface	**Frequency (#)**
Table	7
Wall	11
Combined	18

choose which interface to use, so the samples are not of equal amount. A random sample of n=7, the amount of samples present in the population that chose the *Table* interface, from the population that chose the *Wall* interface was taken and used for the t-test. The time to complete the task using the *Table* interface was lower (20.84 ± 6.46 seconds) than the time to completion using the *Wall* interface (29.10 ± 19.94). The evidence is not enough ($t(12) = 1.04, p = 0.32 > 0.05$) to support affirming one interface faster than the other, considering a confidence interval of 95%.

A2Q4 - What does the colorized terrain mean? For this question, the participants were asked to described what they think the terrain colorization meant. Most testers (17) understood the visualization was meant to convey some characteristic about the wind, with only one user not understanding what the colorization meant. There was general confusion regarding what exact characteristic was being colored, a majority of participants (10), reading the legend, concluded that it portrayed terrain exposure to the wind, while others did not read it and mentioned the color was related to wind intensity (6), which is not correct as there was no data related to intensity. One user mentioned the colors could be encoding the risk the wind could represent for that particular terrain which is related to how exposed the terrain is to the wind. Almost all participants (17) understood they were controlling the wind direction while using the wind tool.

A2Q5 - What do the red colored areas mean? To this question, 3 users answered wrongly, "Terrain more exposed to the Northern Wind", while the remaining 15 answered correctly, "Terrain more exposed to the Wind Direction".

Observations Some relevant information observed regarding this task was that one user mentioned the connection between the colorization and the slope of the terrain, although this user was already familiar with the concept of slopes being facilitators of the fire's progress. This was not the same user that mentioned the concept of risk in question A2Q4.

A smaller observation was that one user understood what the colorization meant after answering question A2Q5.

5.9.4 Group A: Task A3 - Locate “Amêndoa” and “Cardigos” on the Upper Half of the Map

Task A3 intends to teach the participants the location of two towns relevant for later tasks. It also teaches them how to get more information about points of interest in the map. Data related to task A3 is summarized in Table 5.4. Ability to complete this task was observed and the data is available in Table 5.4a. Time to completion, as well as interface chosen were also observed and noted and their data is available in Table 5.4b and Table 5.4c respectively.

A two-sample t-test was used to determine if the use of interface, *Wall* or *Table*, had a significant impact in participants' completion time. Once again, the participants are free to choose which interface to use, so the samples are not of equal amount. A random sample of n=7, the amount of samples present in the population that chose the *Table* interface, from the population that chose the *Wall* interface was taken and used for the t-test. The time to complete the task using the *Table* interface was higher (29.84 ± 17.73 seconds) than the time to completion using the *Wall* interface (13.78 ± 4.15). A statistically significant difference ($t(7) = 2.33, p = 0.04 < 0.05$) is present, considering a confidence interval of 95%, between using the *Wall* and *Table* interface, with the former being faster than the latter.

Observations Some relevant information observed; One user did not immediately focus their efforts in searching the upper-half portion of the map due to miscommunication and spent sometime searching for both towns on all the map until the question was repeated. Another user had previous knowledge of the region and knew were both towns were located. Four users were confused as to wether use one interface or the other with one user suggesting the buttons that reveal the towns on the *Table* and *Wall* should be unified

Table 5.4: Task A3 - Results

(a) A3Q1 - Able to Complete

Completed	Answer (#)
Yes	17
Yes, with help	1
No	0

(b) A3Q2 - Time to Completion (seconds)

Interface Used	$\bar{x} \pm \sigma$ (s)
Table	29.84 ± 17.73
Wall	14.54 ± 03.65
Combined	20.49 ± 13.33

(c) A3Q3 - Interface Used

Interface	Frequency (#)
Table	7
Wall	11
Combined	18

into only one button. Another user mentioned the name labels for towns are sometimes overwhelming and should be toned down. This is a noted problem throughout tasks that necessitate the *Towns* or *Towns3D* layers active, mentioned multiple times by multiple users.

5.9.5 Group A: Task A4 - Place a Marker Ordering Evacuation of Civilians near "Amêndoa"

This next task introduces the user to the *Marker Tool* that will be further tested later on. It relies on knowledge of the location of "Amêndoa", taught in the task just before. The data regarding task A4 is summarized in Table 5.5. For this task, as before, the participants were observed regarding wether they were able to complete the task, Table 5.5a, and also regarding the time it took them to complete the task, Table 5.5b. In this particular task there was no observation regarding which interface the user chose, as the marker can only be placed using the *Table* interface - the participants were instructed as such.

Table 5.5: Task A4 - Results

(a) A4Q1 - Able to Complete

Completed	**Answer (#)**
Yes	17
Yes, with help	1
No	0

(b) A4Q2 - Time to Completion (seconds)

$\bar{x} \pm \sigma$ **(s)**
34.58 ± 16.36

Observations Observed information during this task is not as relevant as previous observations, but one user placed the marker not too close to "Amêndoa" and decided to finish the task, while two others placed it more than once as the first time was not as close as they wanted it to be. An observation one participant made was that the name labels near "Amêndoa" obstruct correct placement of the marker, few users noticed this as it only affects those that use the *Towns3D* layer to locate and place the marker. But is a problem already referred in previous observations.

5.9.6 Group A: Task A5 - Move Wildfire Playback to Monday

The last task from group A intends to show the user they can control the wildfire playback and also to show that the markers themselves are time-sensitive. The results are available in Table 5.6. For this task, as the previous one, the participants were observed on wether they could complete the task or not, Table 5.6a, and how long it took them to complete it, Table 5.6b.

Observations Some participants mentioned the slider to require precise movement, having to keep pointing to the slider exactly, as opposed to some of the behavior observed in other interfaces, where as long as the cursors starts dragging the slider within bounds

Table 5.6: Task A5 - Results

(a) A5Q1 - Able to Complete

Completed	**Answer (#)**
Yes	18
Yes, with help	0
No	0

(b) A5Q2 - Time to Completion (seconds)

$\bar{x} \pm \sigma$ **(s)**
13.26 ± 5.84

it does not really matter if then the cursor moves out of the correct place. This was a noted problem.

5.9.7 Group B - Layer Activation

Group B aims to test which interface the users prefer and which is more efficient. The action the user makes is the same between interfaces, meaning the user uses the *Wall* interface to *activate* two layers and then uses the *Table* interface to *deactivate* the same layers. Both actions are executed in the same way for each interface. Data related to these tasks is presented in Table 5.7 and in a Figure 5.2.

First time the users tried each method in this task there was a difference in completion times as observed, in Table 5.7a, this difference shortened by the second time the participants executed the task. The colors are highlighted as their are later plotted in Figure 5.2. The perceived feeling of comfort had a similar result, with users having a better feeling of comfort by the second time they executed the task, Table 5.7b.

A paired-sample t-test was used to determine if the use of interface, *Wall* or *Table*, had a significant impact in participants' completion time. All participants' times were used in this t-test (n=18) and only the times related to the second execution of the task were considered. The time to complete the task using the *Table* interface was higher (3.16±1.36 seconds) than the time to completion using the *Wall* interface (2.85 ± 0.82). No evidence is present, ($t(17) = 0.86, p = 0.40 > 0.05$), that allows for a distinction of both interfaces with regards to speed, considering a confidence interval of 95%.

Interface Preferences and Observations At the end of the task group, the users were asked which interface they preferred to use, 39% of the participants showed enthusiasm for the *Table* interface while the remaining 61% mentioned the *Wall* interface as their

Table 5.7: Task Group B - Results

(a) BQ1 - Time to Completion (seconds)

Interface	$\bar{x} \pm \sigma$ **(s)**
Wall (1st Time)	2.90 ± 0.90
Wall (2nd Time)	2.85 ± 0.82
Table (1st Time)	6.18 ± 5.40
Table (2nd Time)	3.16 ± 1.36

(b) BQ2 - "I considered it comfortable"

Interface	**Med.**	**1st Q.**	**3rd Q.**
Wall (1st Time)	5	-0	+0
Wall (2nd Time)	5	-0	+0
Table (1st Time)	5	-1.75	+0
Table (2nd Time)	5	-0.75	+0

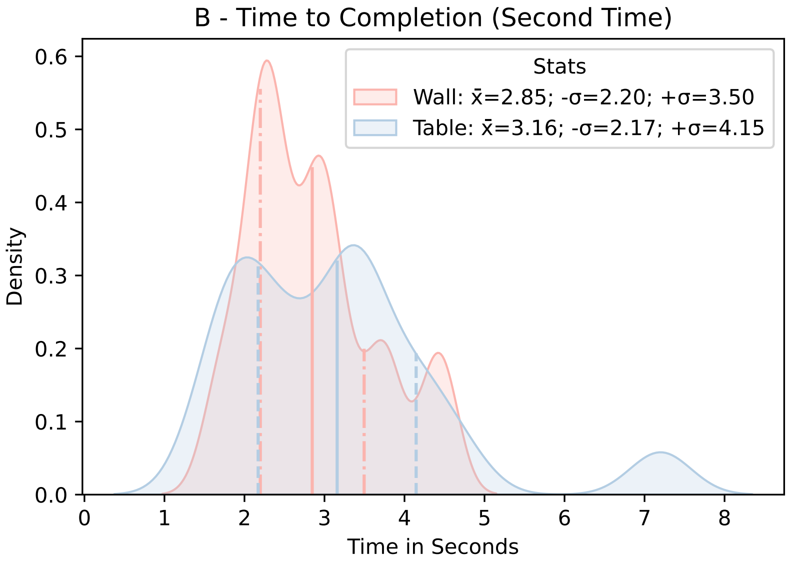


Figure 5.2: Distribution of participants' completion times, upon executing the task a second time with each interface.

method of choice. The reasons expressed for these preferences varied. Some participants preferred the *Table* interface as it is more "physical" and a new way to press buttons, comparing to the usual way of pointing and clicking, much alike the use of the common computer mouse. Other users preferred the *Wall* interface as it covers more real estate in the participants field-of-view, thus being more present both visually and also cognitively.

5.9.8 Group C - Notification System

Task group C tests the different ways available to notify users of new information in the system. The three methods already mention were tested, *Arrow (A)*, *Line (L)* and *Sphere (S)*, by all participants. To avoid biasing the data too much latin square order was used, first participant tested *ALS*, second participant tested *SAL*, third participant tested *LSA*. This pattern repeated throughout the participants.

Data related to this question group is available in Figure 5.3, Figure 5.4 and Table 5.8. Question *CQ2* is omitted as all participants were able to find the marker across all notification types. Figure 5.3 refers to wether the user understood what the notification was and Figure 5.4 summarizes the time it took each participant to find and accept/refuse the marker, captured automatically. Tables 5.8a and 5.8b pertain to the Likert-scale questions present at the end of this group.

In order to analyse wether the completion times of each notification type have a statistically significant difference a one-way repeated measures ANOVA test was applied on the left column data from Figure 5.4. The test revealed that there is no statistically significant difference in completion times when using one notification type over the others. Thus, there is no evidence to reject the null hypothesis ($F(2,34) = 0.99, p = 0.38 >$

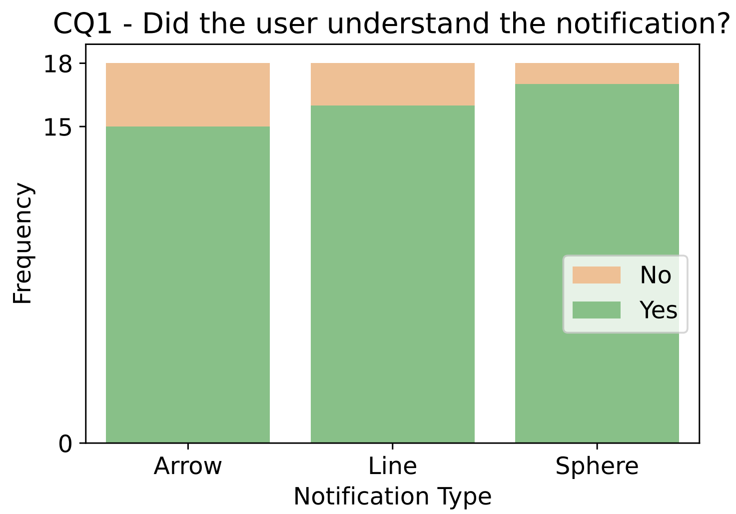


Figure 5.3: Notification Understanding.

0.05) which leads to an inability to chose one notification type over the others with a confidence interval of 95%.

A comparison between each notification type is available in Figure 5.4. In the left column of that figure, the distribution of completion times can be consulted. The right column, visualizes a participants orientation towards the placed marker over time. The y-axis plots the dot product between the participants' forward direction and the participants' direction to the marker, both of these directions are normalized, thus the dot product gives a value in the range of $[-1, 1]$ depending on the participant being with their backs to the marker, -1, perpendicular to the marker, 0, or facing the marker, 1. What is interesting to compare between notification types is how fast the dot product approaches 1, meaning the user is looking in the direction of the placed marker. Each row of this figure corresponds to one such notification type. Other interesting point worth observing is comparing the completion times with user preference, in Figure 5.4a. Even though, the overall faster notification type according to the data is the *Arrow* type, participants largely preferred the *Line* notification type over the *Arrow* type, with the former being considered a first option by twelve participants and the latter by only two. A possible explanation for this might be that the *Arrow* notification is unintrusive enough that users barely lose any time looking at it and instead look to the marker right away. Some further testing should be conducted, particularly with considerably more markers placed around

Table 5.8: Task Group C - Results

(a) CQ4 - I found the marker easily.

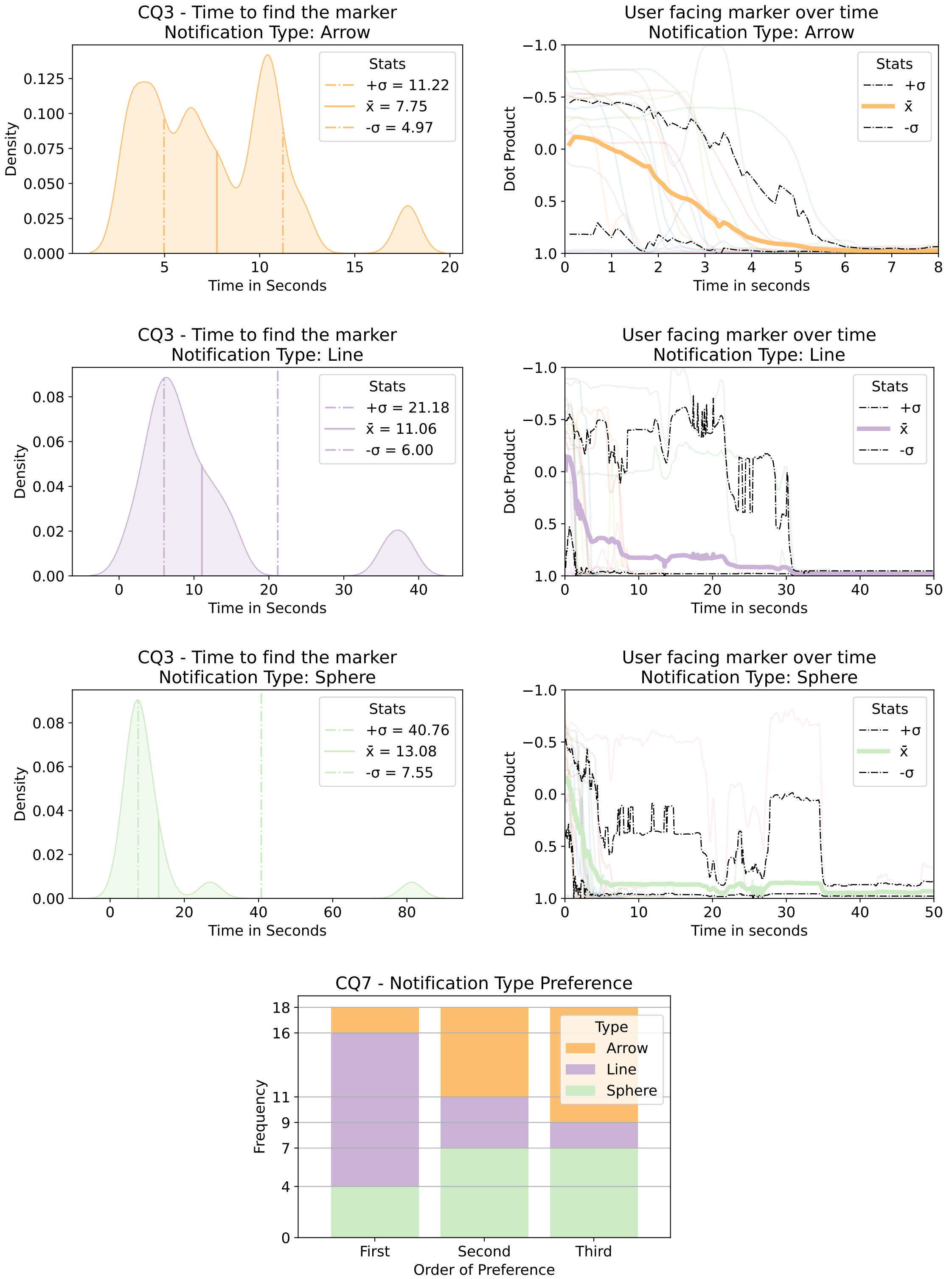
Type	Med.	1st Q.	3rd Q.
Arrow	5	-0	+0
Line	5	-0	+0
Sphere	5	-1	+0

(b) CQ5 - The notification is visible.

Type	Med.	1st Q.	3rd Q.
Arrow	5	-0.75	+0
Line	5	-0.75	+0
Sphere	5	-2	+0

the map so the participant is forced to spend some time figuring out which exact marker the notification is referring to.

Observations During these tests, some participants did not understand they had to start by looking to the controller that vibrated in their hands. This should be rectified making that requirement unnecessary. Four participants mentioned that the *Sphere* notification type was overwhelming and should somehow be toned down, as to not take over the entire field of view. No comments regarding the other two notification types.



(a) Participants' preference.

Figure 5.4: Task C - Completion time distribution alongside participants' orientation towards placed marker over time, followed by participants' preference of notification type.

5.9.9 Group D - Ruler Tool

Group D revolves around the *Ruler Tool*. Participants are asked to take measurements between points of interest, using both *Wall* and *Table* interfaces. The order in which the interfaces were used was alternated for participant to participant but the order of the measurements to be taken was maintained, every user first measures the distance between Cardigos and the Red vehicle and after, the distance between Amêndoa and the Blue vehicle, regardless of interface.

Data related to this task group is presented in Table 5.9, with time to completion presented in Table 5.9a, as well as the participants' measurement and absolute error, ($Error = |UserValue - TrueValue|$), in Table 5.9b.

- Measurement (Msrm.): CR - between Cardigos and Red vehicle = 21.08 Km
- Measurement (Msrm.): AB - between Amêndoa and Blue vehicle = 25.70 Km

Table 5.9: Task Group D - Results

(a) DQ1 - Time to Completion (seconds)

Inter.	Msrm.	$\bar{x} \pm \sigma$ (s)
Wall	CR	17.78 ± 11.26
Wall	AB	10.36 ± 04.90
Table	CR	24.87 ± 16.01
Table	AB	16.31 ± 12.19

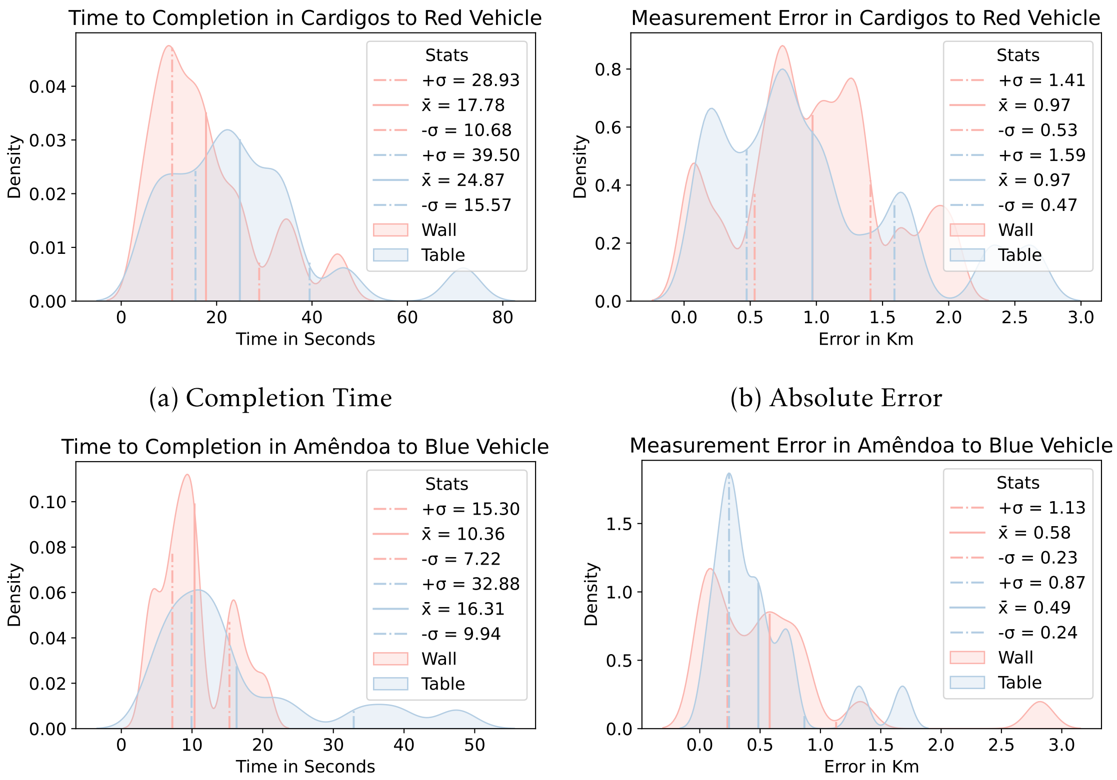
(b) DQ2 - Measurement Value (Km)

Inter.	Msrm.	$\bar{x} \pm \sigma$ (s)	Abs. Error (km)
Wall	CR	20.12 ± 0.58	0.97 ± 0.56
Wall	AB	25.48 ± 0.86	0.57 ± 0.67
Table	CR	21.03 ± 1.23	0.97 ± 0.72
Table	AB	25.62 ± 0.65	0.49 ± 0.42

Figure 5.5 shows a better comparison of completion times, Figures 5.5a and 5.5c, and absolute error, Figures 5.5b and 5.5d, across interfaces and in different measurements. It can be observed that for the second measurement, the bottom row, both metrics improve, as participants understand the exercise.

To determine if there's a statistically significant difference between using the *Wall* or *Table* interfaces to take arbitrary measurements two paired-sample t-tests were conducted, over the errors and completion times, using only data related to the second measurement for both interfaces, the distance between "Amêndoa" and the Blue vehicle.

The time to complete the measurement using the *Wall* interface was lower (10.36±4.90 seconds) than the time to completion using the *Table* interface (16.31±12.19). No evidence, ($t(17) = 0.52, p = 0.61 > 0.05$), points to one interface being faster to use than the other,



(a) Completion Time (b) Absolute Error

(c) Completion Time (d) Absolute Error

Figure 5.5: Task D - Comparison of completion times and absolute error across interfaces.

thus no guarantees can be made about which interface is faster to take measurements, considering a confidence interval of 95%.

The absolute error the participants made using the *Wall* interface was higher (0.58 ± 0.67 Km) than the absolute error using the *Table* interface (0.49 ± 0.42). There is evidence to conclude that one interface might be more precise than the other.There is a statistically significant difference, ($t(17) = 2.35, p = 0.03 < 0.05$), considering a confidence interval of 95%, between using one interface over the other. Particularly, to minimize the error of a given measurement, the *Table* interface is preferred.

User reported comfort was also recorded and is present in Table 5.10.

Interface Preferences and Observations At the end of the task group, the users were asked which interface they preferred to use, 28% of the participants preferred the *Table*

Table 5.10: DQ3 - "I Considered it Comfortable"

Interface	Msrm.	Med.	1st Q.	3rd Q.
Wall	CR	4	-0	+1
Wall	AB	4	-0	+1
Table	CR	4	-0	+1
Table	AB	3.25	-0.75	+0.75

interface while the remaining 72% remarked the *Wall* interface as their method of choice.

For both interfaces, participants expressed difficulty in correctly placing a measuring point due to the amount of visual real estate the name labels take up, this was already mention in the task group A. Some users mentioned preference for the *Table* because they could have the controller very close to the map, making it easier to place the measuring point. Users that preferred the *Wall* interface mentioned that being far away from the interface made it so the small amount of movement their hands made when squeezing the trigger would have a bigger impact their precision.

A small number of participants mentioned an interesting improvement to the tool worth exploring further; when the *Ruler* tool is selected, it could snap to the point-of-interest closest to where the user is pointing, reducing the potential error to near zero.

5.9.10 Group E - Brush Tool

Tasks in group E are meant to test the *Brush Tool*. It's one of the smaller groups with only two tasks, one task where the user must identify an highlighted town followed by a second task where it is the participant that must highlight a town.

Task E1 - Identify the Highlighted Town In this task the researcher highlights the region capital, Mação, in the lower part of the map. The participants are observed if they can correctly identify the town and how fast they can do it.

All participants in this task were able to identify the highlighted town. Participants took on average 8.59 ± 5.73 seconds to identify the town.

Task E2 - Highlight a Town For the second task in group E, the participants were asked to highlight a town using the *Brush Tool*. The participants are free to choose which town to highlight, leading to high variability on completion times, depending on wether the participant highlights a random town, or spends some time looking for a specific one.

All participants were able to complete the task. The interface used to highlight a town was also noted and that data is presented in Table 5.11. In order to understand if there any significant difference between using one interface or the other a two-sample Welch t-test was conducted, since the population size is uneven. Comparing the completion times of participants that used the table interface (13.11 ± 7.83 seconds) with the higher times from participants using the wall interface (16.01 ± 13.04 seconds), the test reports inconclusive results ($t(15) = 0.58, p = 0.56 > 0.05$) leading to a lack of affirmation regarding one interface being better than the other for this particular task.

Table 5.11: Task E2 - Time to Completion (seconds) and interface choice frequency.

Interface	$\bar{x} \pm \sigma$ **(s)**	**Frequency (#)**
Table	13.11 ± 07.83	8
Wall	16.01 ± 13.04	10
Combined	14.72 ± 10.84	18

5.9.11 Group F - Supervisor and Subordinate Scenario

Group F is more of a series of steps rather than discrete tasks. Described earlier, in Section 5.7, this is a collaborative scenario where both users interact with each other to achieve the goal of sending the firetruck to a specific place. For this long task, the participants were observed wether they finished the task or not, how long it took them and then two Likert-scale questions. This data is presented in Table 5.12.

All participants were able to finish the task, their times are show in Table 5.12a, and the answers to the two Liker-scale questions; *FQ2 "I Understood what was Asked"* and *FQ3 - "I Understood I was sharing the VR with another Person"* are available in Table 5.12b.

Table 5.12: Task F - Results

(a) FQ2 - Time to Completion (seconds)

Time to Completion (s)
37.93 ± 17.79

(b) Answers to Likert-scale Questions

Question	**Med.**	**1st Q.**	**3rd Q.**
FQ2	5	-1	+0
FQ3	4.5	-2.25	+0.5

Observations The last step in this task generated quite some confusion for the users. They needed to select a vehicle by hoovering their hands over it and then clicking on the map to order the vehicle to move. The selection motion was very unintuitive, specially taking into account that all, except one, tool related action up to this point were point and click. Almost all participants mentioned that, referring they were expecting to select the vehicle by pointing to it, with the *Vehicle Tool* selected, and clicking to select the vehicle. This was an implementation oversight, as that behavior would be easily implemented. Adding to the confusion, some users also thought they should stay hoovering their hands on the vehicle while trying to click on the vehicle's destination.

5.9.12 System Usability Scale (SUS) Questionnaire

This section discusses the SUS questionnaire results. The data is available in Table 5.13 and the questions can be consulted in Annex I, here the questions have been shortened. The boxplots[2] of the answers are illustrated in Figure 5.6, grouped in an easier way to get a better sense of the result, with the odd questions in Figure 5.6a and even questions in Figure 5.6b.

A quick way to have an idea of how the score will be is to keep in mind that the higher the answers on odd-numbered questions the better and the lower the answers on even-numbered questions the better. This follows from John Brook [5] mentioning how to calculate the final score; odd-numbered answers will be subtracted a value of one and

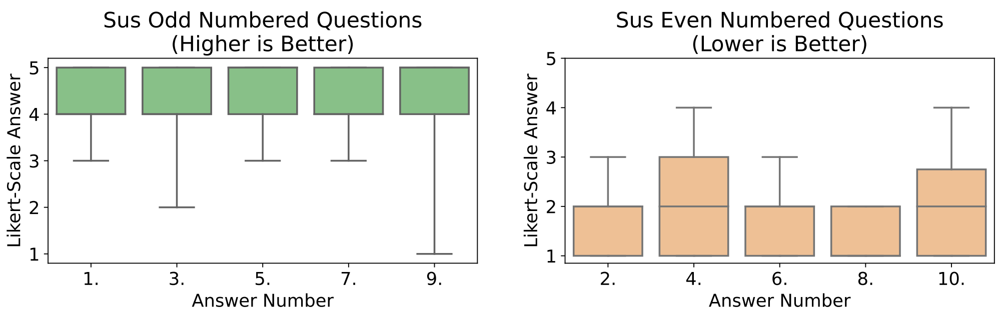
[2]The boxplot visualizes the distribution of the data. The box lower and upper edge represent the 1st and 3rd quartile respectively, while the 2nd quartile (median) is represented by the horizontal bar crossing the box. Sometimes, when the median bar value is not easy to see, a textual label is added. The minimum and maximum values are represented by the lower and upper "whiskers".

even-numbered answers will subtract from five. This normalizes all answers to be in zero to four. Then the sums of each will be multiplied by two and a half to give the final score.

This final scores, illustrated in Figure 5.7, are then compared with the average of 68 points. The results obtained are favorable with only three users giving a score below the average and 83% of users giving a score equal to or above 80 points, leading to a final average score of 82.5 points.

Table 5.13: SUS questionnaire questions shortened and answers. Complete question list available in Annex I.

Questions	Med.	1st Q.	3rd Q.
1. I would use the system frequently.	4	-0	+1
2. The System is complex to use.	2	-1	+0
3. The System is easy to use.	4	-0	+1
4. I would need technical support.	2	-1	+1
5. The system has well integrated functions.	5	-1	+0
6. The system is inconsistent.	2	-1	+0
7. The system is easy to learn.	4	-0	+1
8. The system cumbersome to use.	1	-0	+1
9. I felt confident using the system.	5	-1	+0
10. I needed to learn a lot beforehand.	2	-1	+0.75



(a) Odd-numbered questions.

(b) Even-numbered questions.

Figure 5.6: Boxplot of both even and odd-numbered questions.

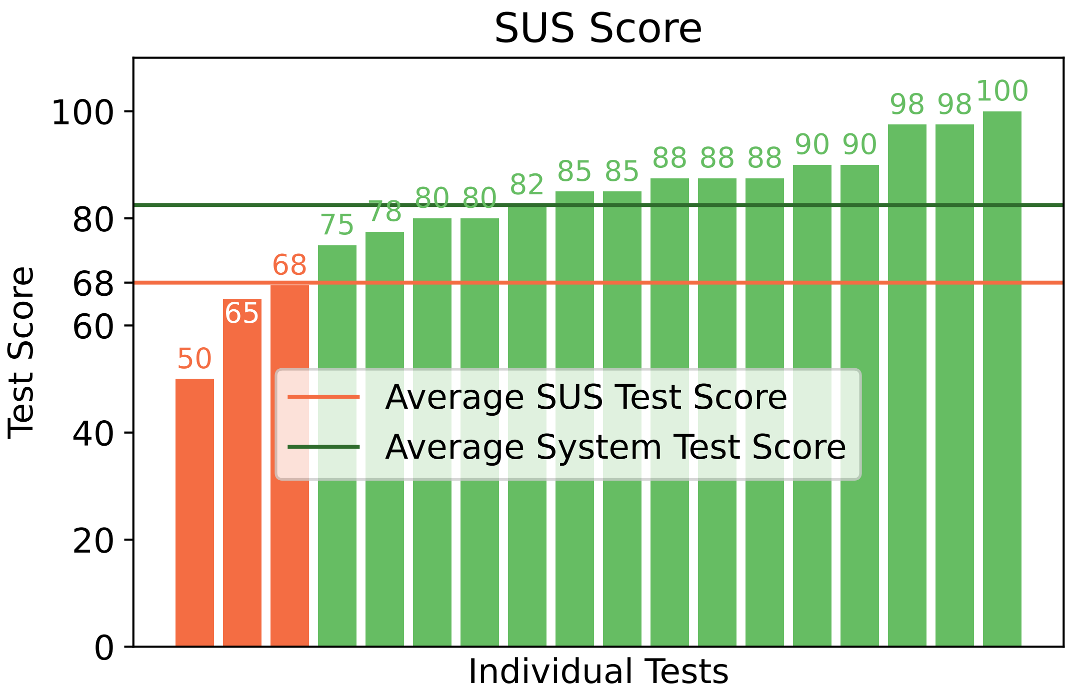


Figure 5.7: Final SUS questionnaire scores. The "Average SUS Test Score" is according to Jeff Sauro, mentioned by John Brook in his article [6]. The "Average System Test Score" is the average score for the system presented in this thesis.

5.9.13 Igroup Presence Questionnaire (IPQ)

This section discusses the IPQ questionnaire results. The data is available in Table 5.14 and the questions can be consulted in annex II, here, as before, the questions have been shortened. The boxplots of the answers are illustrated in Figure 5.8, grouped by the three axis the IPQ aims to test, *Spatial Presence*, *Involvement* and *Experience Realism*, in Figures 5.8a, 5.8b and 5.8c respectively. The *General Item* can be consulted in the first row of Table 5.14.

By observing the boxplots in Figure 5.8, a *Presence Profile* can be made for this particular application. According to the participants, they had high *General* presence, as evidenced by question number 1 (Table 5.14) as well as a high sense of *Spatial Presence* evidenced in Figure 5.8a. The comparatively lower scores in the *Involvement* (Figure 5.8b) questions are expected, as the participants had to be aware of an external person communicating with them from outside the VE, which inhibits participant immersion, having a voice coming from a real world location, rather than a location within the VE. The even lower scores from the *Experienced Realism* (Figure 5.8c) were also expected, as the environment did not strive for photorealism. Overall, the results are consistent with expectations.

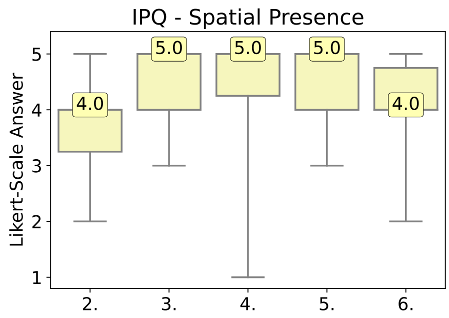
5.9.14 System Questionnaire

Participants answered two system related questions, using a Likert-Scale, these mainly evaluate how the participants would feel using the system to collaborate with another user. The results are resumed in Table 5.15. For convenience sake the questions here were shortened, the full questions can be consulted in Section 5.8.3.

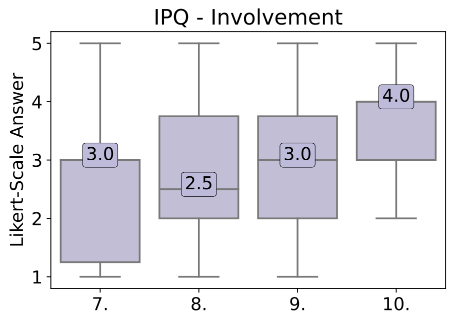
The answers to *SQ1* were better than expected, as the avatar participants were sharing

Table 5.14: IPQ questionnaire questions shortened and answers. For the complete question list, consult Annex II.

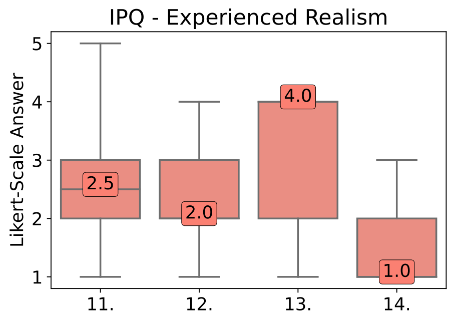
Questions	Med.	1st Q.	3rd Q.
1. I had a sense of "being there".	4.5	-0.5	+0.5
2. The VE surrounded me.	4	-0.75	+0
3. I was just perceiving pictures.	1	-0	+1
4. I did not feel present in the VE.	1	-0	+0.75
5. I had a sense of acting in the VE.	5	-1	+0
6. I felt present in the virtual space.	4	-0	+0.75
7. I was aware of the real world surrounding me.	3	-0	+1.75
8. I was not aware of my real environment.	2.5	-0.5	+1.25
9. I still paid attention to the real environment.	3	-0.75	+1
10. I was completely captivated by the VE.	4	-1	+0
11. How real did the VE seem to you?	2.5	-0.5	+0.5
12. The VE seem consistent with the real world?	2	-0	+1
13. How real did the VE seem to you?	4	-2	+0
14. The VE seemed more realistic than the real world.	1	-0	+1.0



(a) Spatial Presence questions.



(b) Involvement questions.



(c) Experienced Realism questions.

Figure 5.8: Boxplot of the three axis the IPQ tests for.

the VE with was controlled by mouse and keyboard, hence it was static, having none of the organic movements one sees when the avatar is controlled by a VR helmet. Answers to *SQ2* are also very positive, indicating that VR collaboration is a worthy pursuit.

Table 5.15: Answers to the two system related questions.

Questions	**Med.**	**1st Q.**	**3rd Q.**
SQ1 - I was aware I was sharing the VE with someone.	4	-0.75	+1
SQ2 - I could use this system to collaborate with someone.	5	-0	+0

5.10 Discussion

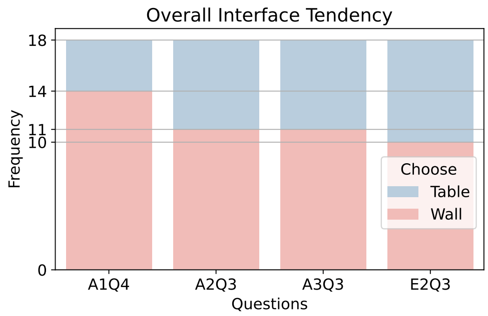
Previous sections were only focused on results within question groups, but this section will focus on results in a more global context, taking into account user characteristics and comparing those to the global task results, it will focus on aspects such as the influence previous experience in virtual reality might have in the completion times of tasks, if it influences interface preference or tendency and other relevant stats. Here, *Preference* is when one participant tries both interfaces and then made his preference known, while *Tendency* is when one participant without any direction from the researcher chose one interface over the other without trying both. For the paragraphs related to VR experience it should be noted that half of the participants involved in the study have previous experience in virtual reality with some having more than others, as reported earlier in Section 5.9.1.

Some insights can be gathered after cross-comparing task data with demographic data. As observed in paragraph **VR experience influence on task performance** below, previous experience in VR doesn't seem to have a statistically significant impact in task performance, this could be interpreted as the natural interactions options the application provides are accessible to both groups of users. The following paragraph **Interface tendency and preference**, indicates participants are more likely to use the more "VR" option of the two as their experience with the system evolves. Drilling down on interface tendency and preference, paragraphs **Interface tendency and VR experience influence on it** and **Interface preference and VR experience influence on it** take a look at how previous VR experience influences those two data points. The visualizations related to those paragraphs indicate that the group with previous experience tended to choose the *Table* interface more frequently than participants without. Interface preference appears to be more dependent on the task itself, indicating there is a place for both interfaces as some tasks are better suited for a two-dimensional interface while others can benefit from a three-dimensional one. Another hypothesis explored was wether a participant's height would influence their tendency/preference to choose a particular interface, paragraph **Participant's height influence on interface tendency and preference** explores this.

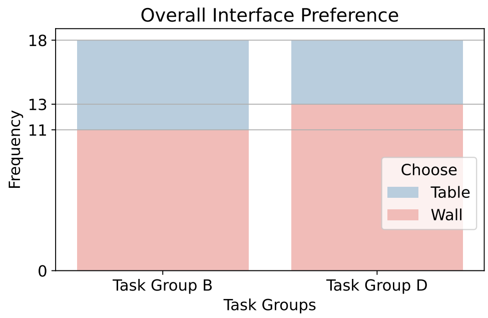
VR experience influence on task performance Table 5.16 compares completion times and the absolute error of measurements across all relevant tasks. In order to understand if previous VR experience has a meaningful influence in task performance, measured as short completion times and low absolute errors, two-sample t-tests were performed for every relevant task. Considering a confidence interval of 95%, there is a lack of evidence indicating that previous VR experience has a statistically significant impact on performance with only one test, *A2Q2*, indicating a statistically significant result.

Table 5.16: Comparison of time to completion and absolute error in measurements across all relevant questions.

Time Questions (s)	**Some VR Exp.**	**No VR Exp.**	**T-Test (CI = 95%)**
A1Q3 - Completion Time	4.49 ± 0.92	5.34 ± 4.18	$t(16) = 0.60, p = 0.56$
A2Q2 - Completion Time	31.93 ± 17.15	19.51 ± 4.72	$t(16) = 2.09, p = 0.05$
A3Q2 - Completion Time	25.11 ± 16.91	15.87 ± 6.62	$t(16) = 1.53, p = 0.14$
A4Q2 - Completion Time	36.12 ± 13.28	33.05 ± 19.68	$t(16) = 0.39, p = 0.70$
A5Q2 - Completion Time	14.17 ± 5.64	12.36 ± 6.24	$t(16) = 0.65, p = 0.53$
B1Q1 - Completion Time	2.93 ± 0.75	2.86 ± 1.06	$t(16) = 0.16, p = 0.88$
B2Q1 - Completion Time	3.13 ± 0.76	2.56 ± 0.81	$t(16) = 1.56, p = 0.14$
B3Q1 - Completion Time	7.12 ± 7.07	5.25 ± 3.13	$t(16) = 0.72, p = 0.48$
B4Q1 - Completion Time	3.43 ± 1.61	2.89 ± 1.10	$t(16) = 0.83, p = 0.42$
C1Q3 - Completion Time	7.39 ± 3.12	8.10 ± 4.66	$t(16) = 0.38, p = 0.71$
C2Q3 - Completion Time	10.01 ± 10.21	12.12 ± 10.68	$t(16) = 0.43, p = 0.67$
C3Q3 - Completion Time	16.82 ± 24.32	9.35 ± 6.98	$t(16) = 0.89, p = 0.39$
D1Q1 - Completion Time	18.44 ± 11.59	17.11 ± 11.59	$t(16) = 0.24, p = 0.81$
D2Q1 - Completion Time	10.34 ± 5.22	10.38 ± 4.86	$t(16) = 0.01, p = 0.99$
D3Q1 - Completion Time	29.18 ± 20.30	20.57 ± 9.54	$t(16) = 1.15, p = 0.27$
D4Q1 - Completion Time	14.73 ± 13.20	17.89 ± 11.66	$t(16) = 0.54, p = 0.60$
E1 - Completion Time	8.60 ± 3.43	8.58 ± 7.62	$t(16) = 0.01, p = 0.99$
E2 - Completion Time	13.56 ± 11.20	15.88 ± 11.01	$t(16) = 0.44, p = 0.66$
F - Completion Time	34.93 ± 15.73	40.93 ± 20.13	$t(16) = 0.71, p = 0.49$
Distance Questions (km)	**Some VR Exp.**	**No VR Exp.**	**T-Test (CI = 95%)**
D1Q2 - Distance (Abs. Er.)	0.99 ± 0.57	0.95 ± 0.58	$t(16) = 0.15, p = 0.88$
D2Q2 - Distance (Abs. Er.)	0.68 ± 0.87	0.48 ± 0.41	$t(16) = 0.62, p = 0.54$
D3Q2 - Distance (Abs. Er.)	1.16 ± 0.73	0.78 ± 0.70	$t(16) = 1.12, p = 0.28$
D4Q2 - Distance (Abs. Er.)	0.58 ± 0.54	0.39 ± 0.26	$t(16) = 0.96, p = 0.35$



(a) Interface Tendency



(b) Interface Preference

Figure 5.9: Interface tendencies and preferences.

Interface tendency and preference Over the course of the session, participants tendencies and preferences regarding the available interfaces were recorded and are visualized in Figure 5.9, with tendencies on Figure 5.9a and preferences in Figure 5.9b. Overall, the *Wall* is the interface the users tend to and also the interface users preferred when asked to choose between the two. This is an expected result, as the *Wall* interface is considered more natural to people used to interfacing with a computer or smartphone using a mouse a touchscreen; the point and click method is pervasive, so it stands to reason that the *Wall* would be the natural tendency most participants have.

It is also interesting to note the evolution on the tendency. On a first interaction with the system only 22% of participants chose the *Table* to complete the task. By the last task, 44% of participants naturally tended to use the *Table* interface. This reflects how participants got more at ease with the system over the course of the session.

Another interesting comparison to make is how participants' tendencies align with their preferences. For this, Confusion Matrices were elaborated comparing all the tendency data, from tasks *A1*, *A2*, *A3* and *E2*, with the preferences stated by the participants upon completing *Group B* and *Group D*. The visualization can be consulted in Figures 5.11. Then those results were consolidated into combined *confusion matrix* showing the averages of tendencies across tasks and preferences across groups. This overall matrix is visualized in Figure 5.10. Observing this matrix it is possible to conclude that the most common case is for a participant to have an initial tendency for the *Wall* interface and later, when prompted with a choice, also prefer that same *Wall* interface; this is observed on average 48% of the time and is expected, as already mentioned, due to the *Wall* interface being similar to the point-and-click interfaces of daily life. It is also interesting to note the percentage of participants that initially tended for one interface, but later one claimed preference for another; 35% of participants exhibited this behavior. Some interesting observations can also be made when evaluating the matrices comparing individual tasks against individual groups. For instance, in Figure 5.11a and 5.11b, there is a very high percentage (61%) of participant's that tended for the *Wall* interface and actually preferred it when asked later. This might be due to the it being very first and the participant

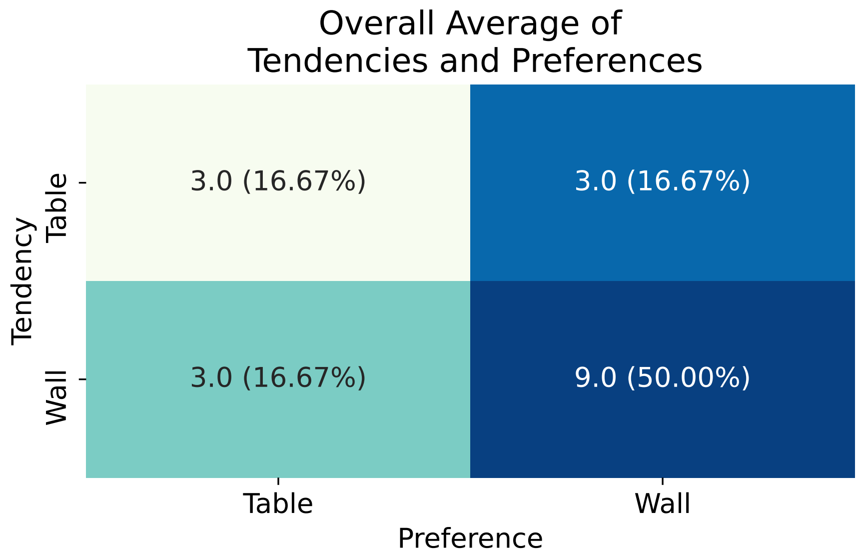


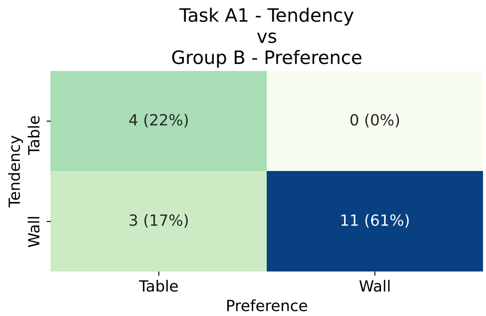
Figure 5.10: Confusion matrix comparing average interface tendency values across all tasks against average interface preference across all groups.

not being fully comfortable with exploring the possibilities in the VE and just goes with the more natural *Wall* interface. Results regarding comparison of task tendency in *Task A3* and preference in *Task Group B* are surprising as a high percentage of participants (56%) conflicting tendencies and preferences. This might be due to *Task A3* being about locating a town by just pointing to it, thus leading participants to experiment with the other interface, as opposed to the one they prefer, more frequently.

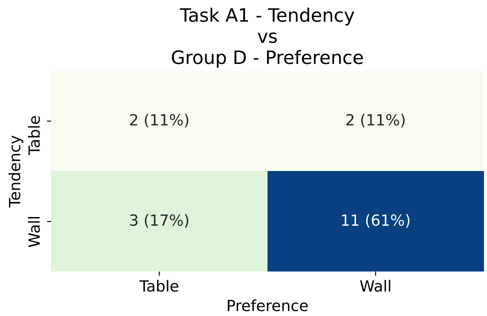
Interface tendency and VR experience influence on it Figure 5.12 presents data related to the influence previous VR experience might have on tendency to pick an interface over another. The data is only related to questions that allowed participants freedom to choose which interface to use, these are; question *A1Q4*, question *A2Q3*, question *A3Q3* and task *E*, presented in Figures 5.12a, 5.12b, 5.12c and 5.12d respectively.

Observing the data visualizations one can see that more participants tend to use the *Wall* interface over the *Table* interface, as stated before, this is no surprise. Although, interestingly enough, over the course of the session the number of people choosing the *Table* interface grows. It is also observable that participants with previous VR experience choosing the *Table* tend to outnumber the ones who didn't in all tasks.

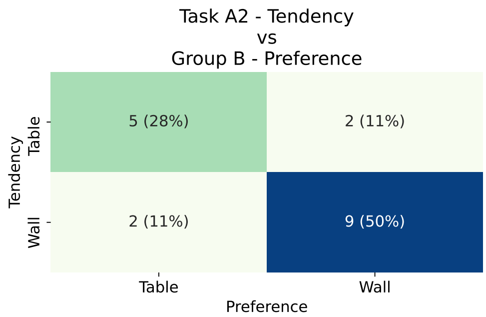
Interface preference and VR experience influence on it Figure 5.13 visualizes data related to the two questions where the participants try both interfaces and then are asked which one they prefer, *Task Group B*, Figure 5.13a and *Task Group C*, Figure 5.13b, and how each group of participants, with or without previous VR experience, answered when asked which interface they preferred for those tasks. VR experience doesn't appear to



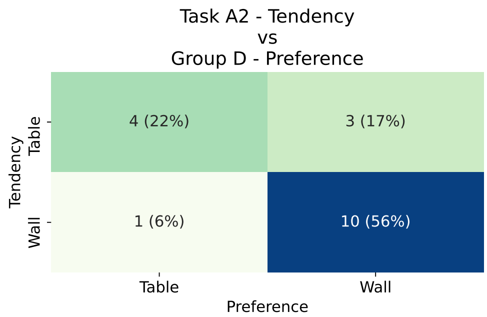
(a) Comparing Task A1 and Group B.



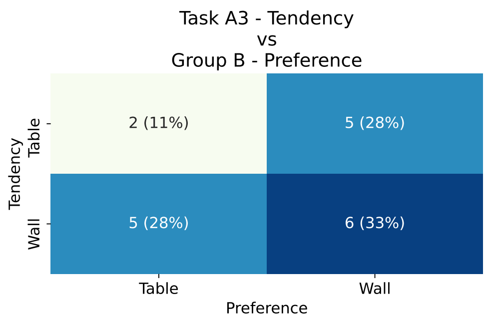
(b) Comparing Task A1 and Group D.



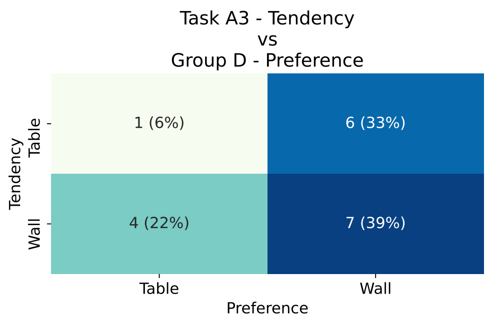
(c) Comparing Task A2 and Group B.



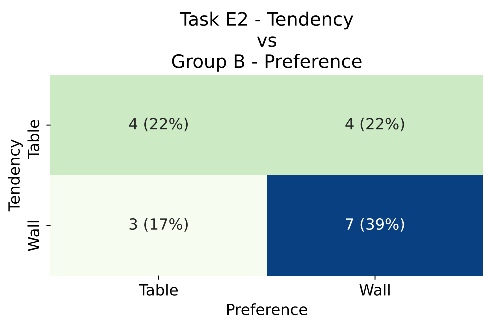
(d) Comparing Task A2 and Group D.



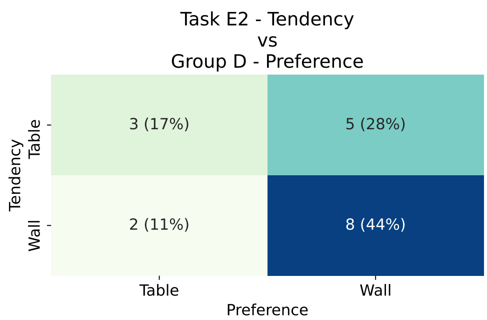
(e) Comparing Task A3 and Group B.



(f) Comparing Task A3 and Group D.



(g) Comparing Task E2 and Group B.



(h) Comparing Task E2 and Group D.

Figure 5.11: Confusion matrices comparing interface tendency against interface preference.

influence participants' preference as much as the task itself, as most participants mentioned, in *Group D*, preferring the *Wall* interface as it is easier for taking measurements even if these measurements are less accurate than the ones taken using the *Table* interface, as previously mentioned in Section 5.9.9.

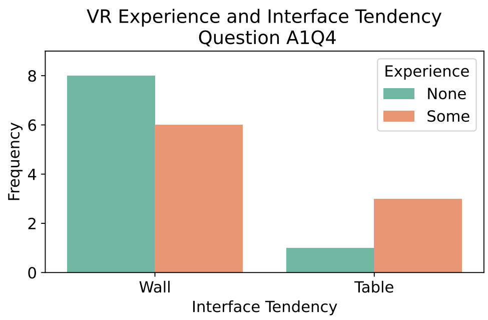
Participant's height influence on interface tendency and preference Participant's tendency or preference of one interface over another could be influenced by the participant's own height, as a taller user can have a better view of the map placed on the *Table*. Figure 5.14 resumes tendency data, the groups were divided by the mean height of the population which was 1.70 ± 0.09 meters, it is also relevant to note that the map on the table was positioned at a height of 0.79 meters, approximately the height of a common table. Figures 5.14a, 5.14b, 5.14c and 5.14d visualize tendency data of questions *A1Q4*, *A2Q3*, *A3Q3* and *E2Q3* respectively. There is no indication that height as an impact on interface tendency. Height influence was also explored and is available in Figure 5.15, with specific task groups *B* and *D* in Figures 5.15a and 5.15b. In this case, it appears that participants with below average height heavily preferred using the *Wall* interface with only one below average height participant showing preference for the *Table* interface.

5.11 Summary

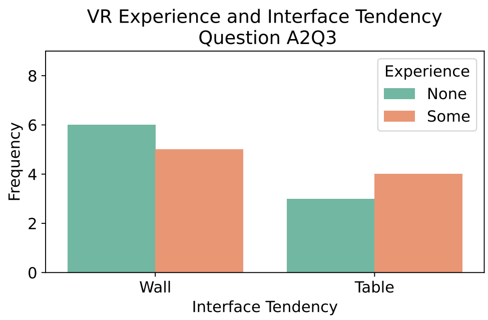
This chapter analysed and discussed data gathered throughout the user testing phase of this thesis. Individual tasks were first discussed on their own terms and by the end they were crossed with some demographic data. Even though the number of participants wasn't abundant some interesting insights were gleaned from the testing phase and mentioned throughout. Another look at the initial research questions should be taken, while some questions and more conclusive answers than others, the work was able to provide general insight into each questions.

- Q - Can a crisis management scenario be improved in a collaborative VR environment?
- Q1 - Are there benefits to a dual-map (2D/3D) visualization?
- Q2 - Is there a preference for one interface over the other?
- Q3 - How can users notify each other of new information in the system?
- Q4 - Are users with no VR experience able to adapt quickly to the VE and the available tools?

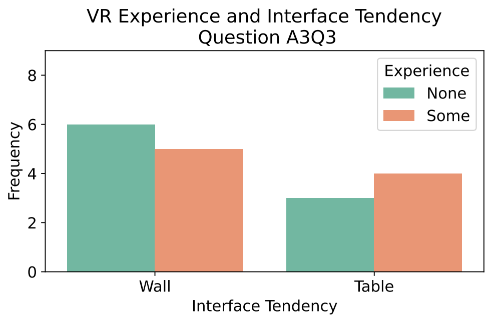
Starting with question *Q1*, data supports that both interfaces are helpful and should be available, evidence when comparing both interfaces in task groups A, B and D, in Sections 5.9.2, 5.3 and 5.9.9. Having both *Wall* and *Table*, interfaces allows participants



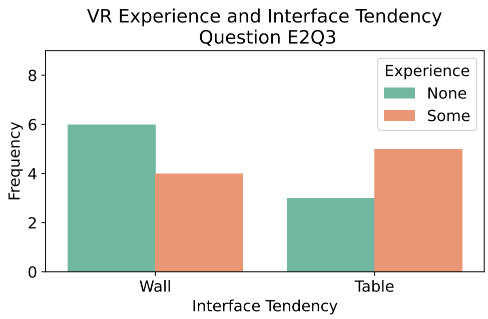
(a) Question A1Q4



(b) Question A2Q3

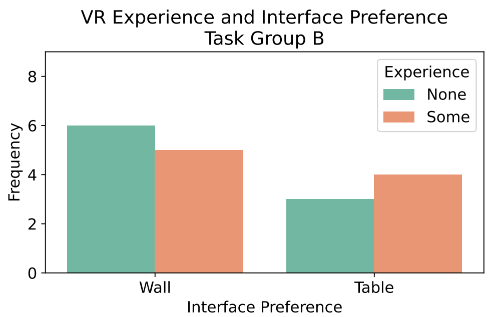


(c) Question A3Q3

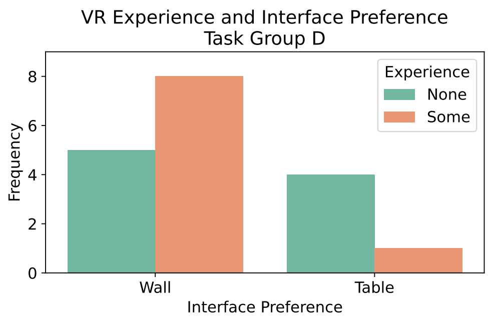


(d) Question E2Q3

Figure 5.12: VR experience and interface tendency.

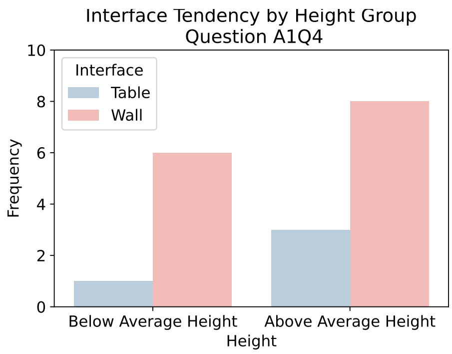


(a) Task Group B

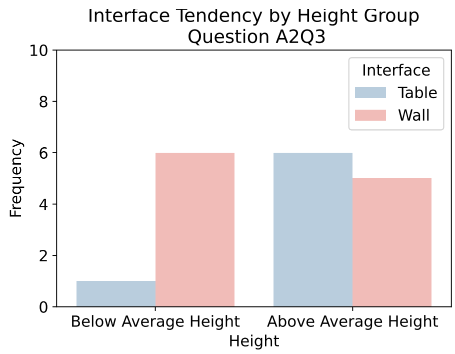


(b) Task Group D

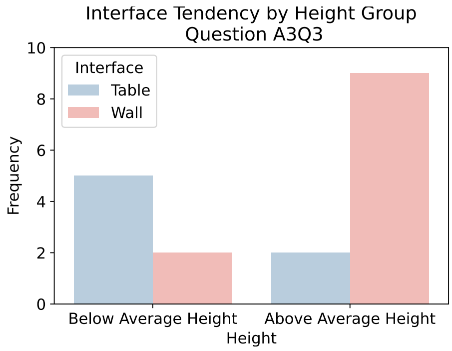
Figure 5.13: VR experience and interface preference.



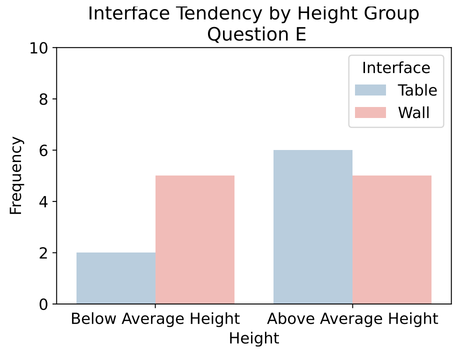
(a) Question A1Q4



(b) Question A2Q3

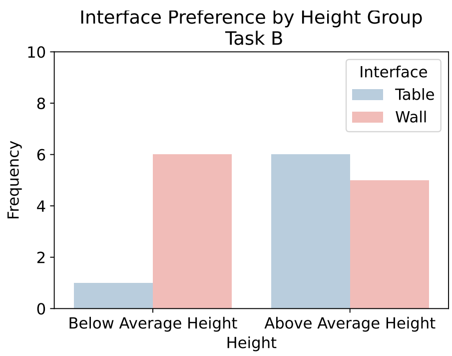


(c) Question A3Q3

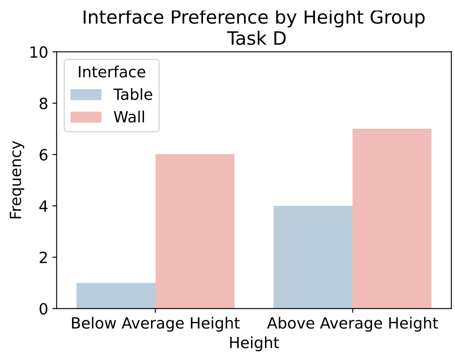


(d) Question E2Q3

Figure 5.14: Interface tendency grouped by participant's height.



(a) Task Group B



(b) Task Group D

Figure 5.15: Interface preference grouped by participant's height.

to choose which interface is better suited for a given tasks; it can also benefit having users interact with each maps while not disturbing each other.

Data regarding question *Q2*, present in paragraph **Interface tendency and preference** in Section 5.10, indicates a preference for the *Wall*, although by a small margin, as it is a more natural way to interact.

For question *Q3* this thesis evaluated only three techniques for notifying users and concluded that time to completion is not a determining factor for user preference. Even though, the ANOVA test did not report statistically significant differences mean completion times, as presented in Section 5.9.8, the *Arrow* notification type is on average faster than both other methods but was consistently placed below the others in order of preference, placing only once in first place, with the *Line* type being the favorite.

As for question *Q4*, it appears to have a positive answer, as participants, both experienced and not in VR, were able to quickly adapt to the system and its collaborative mechanisms and completion times did not have a statistically significant difference, as observed in paragraph **VR experience influence on task performance** in Section 5.10. Any difficulties that arose where quickly dealt with by the participant's own experiments or a quick verbal explanation from the researcher.

Finally, question *Q* lacks a clearer answer, but evidence from the study supports a positive answer. Similar systems may not be a substitution to physical collaboration but a complement to the current methods, allowing for participation from experts not localized in the same physical space and more interactive visualizations of multiple datasets. In order to reach a more conclusive answer, the system should be tested with experts. Due to the Covid-19 limitations, unfortunately, that was unfeasible.

For a more conclusive set of answers, a test session should be made with both the participants and the researcher inside the VE, all using VR headsets, although this was initially planned, as well as an expert testing the system, the Covid-19 limitations in placed made it impossible to execute.

Chapter 6

Conclusion

Virtual Reality (VR) is an exciting field to research in. The technology is more accessible than ever, with smartphones already able to provide immersive VR experiences. The field has been in development for a long time, but it is also, young enough to not have an extensive amount of established conventions which make it prime for exploration of new ways to interact, engage and communicate with and among users.

Collaboration in VR is an interesting subject as it has great potential for exploration, and as such it was the main focus of this thesis. The explored collaboration scenario was a wildfire crisis scenario. Initially, work started by researching collaboration within a Virtual Environment (VE), VR interactions, collaborative and not, applications and health issues related to VR, presented in Chapter 2. Then, a plan to develop a collaborative application where multiple users could connect to a VR and interact with a 2D and 3D representation of Mação was developed over the course of Chapter 3, referring system features and architecture. Following that, Chapter 4 dived into implementation of said features and architecture, explaining system capabilities and elaborating when thought necessary or interesting. Chapter 5 went over testing protocol, tasks and questionnaires presented to participants over testing session and, by the second half, it addressed data collected, results obtained and insights gleamed. Finally, the aptly named Chapter 1, presented the research questions that guided the work, the motivations and expected contributions for this thesis.

Definitive answers are not necessarily easy to find with a limited amount of users testing the system, which is compounded by the inherited complexity of testing with VR hardware and physical space requirements. Furthermore, the Covid-19 pandemic complicated the testing phase, particularly the collaborative tests. Taking into account these restrictions, both hardware and health related, a total of eighteen test sessions were completed, following all the health precautions. These test sessions generated enough data to start forming possible answers to the posed research questions, elaborated in Section 5.11. As mentioned in Section 5.11, data supports a positive answer to question *Q1*, indicating the presence of a dual-map interface to be positive. Data also indicates a small preference for the *Wall* interface, answering question *Q2*. For question *Q3*, participants did not have a statistically significant difference in mean completion times, but there was a preference for the *Line* type of notification. Answering question *Q4* both groups of participants, with and without VR experience, were able to quickly adapt to the VE and the available tools. Finally, the main question, *Q*, lacks a clear answer but evidence from the study supports a positive one. A more definitive answer could be obtained once the system is tested with experts, due to Covid-19 limitations this was an impossibility.

Developing a Master's Thesis during Covid-19 Pandemic It's impossible to avoid the impact Covid-19 had and still has on the world. This thesis was entirely developed while the global pandemic posed a constant risk to everyone. I was fortunate enough to be able to develop most of my work while staying safe at home, only visiting the University when hardware was necessary or during the testing sessions, which were done with utmost care to minimize risk of transmission as much as possible. Developing a thesis on a particular subject such as VR from home posed some difficulties, particularly related to feedback loops between my advisors and I. The testing phase was also severely impacted; during planning the test sessions were thought as having two participants both using head-mounted displays testing the system, quickly it became obvious this would not be allowed under Covid-19 restrictions, rightfully so, so it had to be adapted to the version presented in Section 5.1. The restrictions imposed also limited the amount of test sessions that could occur on a daily basis imposing another limiting factor to the number of participants in the user study.

6.1 Future Work

While developing the system and during user testing sessions some ideas for future research directions and overall improvements to the application popped up or were suggested by the participants. Some of these ideas will be mentioned below.

Networking Probably the weakest link in the system is the networking element. Networking not being the focus of the thesis allowed for some avoidance of the topic, but when it comes to collaboration is an absolute necessity. The current implementation is workable, but, in order to build upon it, some tools should be redesigned with the newly acquired knowledge, particularly, the *Brush Tool*; as it became a network bottleneck once the testing was done through Wi-Fi instead of a wired connection. Other important network problem that should be prioritized is making it possible to collaborate over the internet and not just only over local network, current implementation should allow for that but no work towards it was invested. Both these problems were a non-priority during development as the lab were testing took place limited external connections, thus encouraging a local network based solution.

Prototype and Tools There is always something that could be improved, but the more demanding aspects would be related to geographical data import, as the current methods are specific for the data that was needed, one could use the exact same methods to import data related to another region and that would work, but it does not support the full GeoJSON specification which sooner or later will come up. Some good visual design would also improve the VE's look considerably. Taking some suggestions from the testing sessions, the *Notification System* should not require the user to look to the controller before showing the notification, it should be presented as soon as it is available, but still in a first-in-first-out fashion, as some users were observed completely ignoring the controller and then not noticing the notification visual. The *Ruler Tool* would benefit immensely from a snapping mechanism that would pick points of interest near the measuring point, potentially reducing the error rate to near zero, as a few users suggested. Finally, the *Vehicle Tool* should follow the same way to interact as other tools, as most users thought they would need to point and click to a vehicle in order to select it which was not the case.

Further Research A very interesting test would be to have both users using VR equipment, which the application supports but testing of it was impossible due to both hardware and Covid-19 limitations. Interesting research might be found in exploring direct interaction between the VR user and the geographical data loaded in the VE, allowing the user, for instance, to mark roadblocks, field agents and other resources. This would benefit from a real-time updating mechanism instead of the current solution which loads everything once and makes that data immutable. Adding to that, another interesting avenue to research would be future prospects for the scenario based on projections and previsions from meteorology and other sources. One last interesting, but maybe counter-intuitive, avenue to investigate would be to make it so that each map has "private" views so various users could work or compose visualizations in their own view and then toggle those views on and off, comparing them between users.

Bibliography

[1] D. Ardal, S. Alexandersson, M. Lempert, and A. T. Abelho Pereira. "A Collaborative Previsualization Tool for Filmmaking in Virtual Reality." In: *European Conference on Visual Media Production*. CVMP '19. New York, NY, USA: Association for Computing Machinery, 2019. ISBN: 9781450370035. DOI: 10.1145/3359998.3369404.

[2] T. Azai, M. Otsuki, F. Shibata, and A. Kimura. "Open Palm Menu: A Virtual Menu Placed in Front of the Palm." In: *Proceedings of the 9th Augmented Human International Conference*. AH '18. New York, NY, USA: Association for Computing Machinery, 2018. ISBN: 9781450354158. DOI: 10.1145/3174910.3174929.

[3] T. Azai, S. Ushiro, J. Li, M. Otsuki, F. Shibata, and A. Kimura. "Tap-Tap Menu: Body Touching for Virtual Interactive Menus." In: *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*. VRST '18. New York, NY, USA: Association for Computing Machinery, 2018. ISBN: 9781450360869. DOI: 10.1145/3281505.3281561.

[4] R. M. Baños, V. Guillen, S. Quero, A. García-Palacios, M. Alcaniz, and C. Botella. "A Virtual Reality System for the Treatment of Stress-Related Disorders: A Preliminary Analysis of Efficacy Compared to a Standard Cognitive Behavioral Program." In: *Int. J. Hum.-Comput. Stud.* 69.9 (2011), pp. 602–613. ISSN: 1071-5819. DOI: 10.1016/j.ijhcs.2011.06.002.

[5] J. Brooke. "SUS: A quick and dirty usability scale." In: *Usability Eval. Ind.* 189 (Nov. 1995).

[6] J. Brooke. "SUS: A Retrospective." In: *J. Usability Studies* 8.2 (Feb. 2013), 29–40. ISSN: 1931-3357.

[7] M. Cavallo, M. Dolakia, M. Havlena, K. Ocheltree, and M. Podlaseck. "Immersive Insights: A Hybrid Analytics System for Collaborative Exploratory Data Analysis." In: *25th ACM Symposium on Virtual Reality Software and Technology* (2019). DOI: 10.1145/3359996.3364242.

[8] D. Coffey, N. Malbraaten, T. Le, I. Borazjani, F. Sotiropoulos, and D. F. Keefe. "Slice WIM: A Multi-Surface, Multi-Touch Interface for Overview+detail Exploration of Volume Datasets in Virtual Reality." In: *Symposium on Interactive 3D Graphics and Games*. I3D '11. San Francisco, California: Association for Computing Machinery, 2011, 191–198. ISBN: 9781450305655. DOI: 10.1145/1944745.1944777.

[9] A. Conges, A. Evain, F. Benaben, O. Chabiron, and S. Rebiere. "Crisis Management Exercises in Virtual Reality." In: *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 2020, pp. 87–92. ISBN: 978-1-7281-6532-5. DOI: 10.1109/VRW50115.2020.00022.

[10] A. Conges, P. Yang, F. Benaben, and C. J. Graham. "Using Virtual Reality to Facilitate Common Operational Pictures' Representation." In: *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 2021, pp. 336–341. DOI: 10.1109/VRW52623.2021.00067.

[11] S. Doeweling, T. Tahiri, P. Sowinski, B. Schmidt, and M. Khalilbeigi. "Support for Collaborative Situation Analysis and Planning in Crisis Management Teams Using Interactive Tabletops." In: *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*. ITS '13. St. Andrews, Scotland, United Kingdom: Association for Computing Machinery, 2013, 273–282. ISBN: 9781450322713. DOI: 10.1145/2512349.2512823.

[12] C. Elvezio, M. Sukan, O. Oda, S. Feiner, and B. Tversky. "Remote Collaboration in AR and VR Using Virtual Replicas." In: *ACM SIGGRAPH 2017 VR Village*. SIGGRAPH '17. New York, NY, USA: Association for Computing Machinery, 2017. ISBN: 9781450350136. DOI: 10.1145/3089269.3089281.

[13] C. Elvezio, F. Ling, J.-S. Liu, and S. Feiner. "Collaborative Virtual Reality for Low-Latency Interaction." In: *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings*. UIST '18 Adjunct. New York, NY, USA: Association for Computing Machinery, 2018, pp. 179–181. ISBN: 9781450359498. DOI: 10.1145/3266037.3271643.

[14] C. Elvezio, M. Sukan, and S. Feiner. "Mercury: A Messaging Framework for Modular UI Components." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1–12. ISBN: 9781450356206. DOI: 10.1145/3173574.3174162.

[15] J. Gipson, L. Brown, E. Robbins, J. Gomez, M. Anderson, J. Velasquez, J. Ruiz, and D. Cooper. "VR Story Production on Disney Animation's "Cycles"." In: *ACM SIGGRAPH 2018 Talks*. SIGGRAPH '18. New York, NY, USA: Association for Computing Machinery, 2018. ISBN: 9781450358200. DOI: 10.1145/3214745.3214818.

[16] J. Gonçalves and J. C. Fernandes. "Assessment of SRTM-3 DEM in Portugal With Topographic Map Data. Workshop: 3D Remote Sensing - Use Of The Third Dimension For Remote Sensing Purposes June 10th – 11th 2005." In: *In Earsel Symposium*. 2005.

[17] P. Heidicker, E. Langbehn, and F. Steinicke. "Influence of avatar appearance on presence in social VR." In: *2017 IEEE Symposium on 3D User Interfaces, 3DUI 2017 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., 2017, pp. 233–234. ISBN: 9781509067169. DOI: 10.1109/3DUI.2017.7893357.

[18] H. Ibayashi, Y. Sugiura, D. Sakamoto, N. Miyata, M. Tada, T. Okuma, T. Kurata, M. Mochimaru, and T. Igarashi. "Dollhouse VR." In: *SIGGRAPH Asia 2015 Emerging Technologies on - SA '15*. SA '15. New York, New York, USA: ACM Press, 2015, pp. 1–2. ISBN: 9781450339254. DOI: 10.1145/2818466.2818480.

[19] International Society for Presence Research. *The Concept of Presence: Explication Statement*.

[20] J. Jerald. *The VR Book: Human-Centered Design for Virtual Reality*. Association for Computing Machinery and Morgan & Claypool, 2015. ISBN: 9781970001129.

[21] D. M. Johnson. *Introduction to and Review of Simulator Sickness Research*. Tech. rep. US. Army Research Institute for the Behavioral and Social Sciences, 2005.

[22] D. P. Käser, E. Parker, A. Glazier, M. Podwal, M. Seegmiller, C.-P. Wang, P. Karlsson, N. Ashkenazi, J. Kim, A. Le, M. Bühlmann, and J. Moshier. "The Making of Google Earth VR." In: *ACM SIGGRAPH 2017 Talks*. SIGGRAPH '17. New York, NY, USA: Association for Computing Machinery, 2017. ISBN: 9781450350082. DOI: 10.1145/3084363.3085094.

[23] J. Knispel and F. Bullock. "Collaborative VR Painting in Web Browsers." In: *SIGGRAPH Asia 2017 VR Showcase*. SA '17. New York, NY, USA: Association for Computing Machinery, 2017. ISBN: 9781450354080. DOI: 10.1145/3139468.3148451.

[24] W. S. Lages, M. Nabiyouni, and L. Arantes. "Krinkle cube A collaborative VR game using natural interaction." In: *CHI PLAY 2016 - Proceedings of the Annual Symposium on Computer-Human Interaction in Play Companion*. Association for Computing Machinery, Inc, 2016, pp. 189–196. ISBN: 9781450344586. DOI: 10.1145/2968120.2987746.

[25] I. Lediaveva and J. LaViola. "Evaluation of Body-Referenced Graphical Menus in Virtual Environments." In: *Graphics Interface 2020 Conference* (2020).

[26] G. Llorach, A. Evans, and J. Blat. "Simulator Sickness and Presence Using HMDs: Comparing Use of a Game Controller and a Position Estimation System." In: *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*. VRST '14. New York, NY, USA: Association for Computing Machinery, 2014, pp. 137–140. ISBN: 9781450332538. DOI: 10.1145/2671015.2671120.

[27] D. Luxen and C. Vetter. "Real-time routing with OpenStreetMap data." In: *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. GIS '11. Chicago, Illinois: ACM, 2011, pp. 513–516. ISBN: 978-1-4503-1031-4. DOI: 10.1145/2093973.2094062.

[28] E. Makled, A. Yassien, P. Elagroudy, M. Magdy, S. Abdennadher, and N. Hamdi. "PathoGenius VR: VR Medical Training." In: *Proceedings of the 8th ACM International Symposium on Pervasive Displays*. PerDis '19. New York, NY, USA: Association for Computing Machinery, 2019. ISBN: 9781450367516. DOI: 10.1145/3321335.3329694.

[29] D. Medeiros, E. Ribeiro, P. Dam, R. Pinheiro, T. Motta, M. Loaiza, and A. B. Raposo. "A Case Study on the Implementation of the 3C Collaboration Model in Virtual Environments." In: *2012 14th Symposium on Virtual and Augmented Reality*. 2012, pp. 147–154. DOI: 10.1109/SVR.2012.28.

[30] R. Nóbrega, J. Jacob, R. Rodrigues, A. Coelho, and A. A. d. Sousa. "Augmenting Physical Maps: an AR Platform for Geographical Information Visualization." In: *EG 2016 - Posters*. Ed. by L. G. Magalhaes and R. Mantiuk. The Eurographics Association, 2016. DOI: 10.2312/egp.20161052.

[31] G. Papagiannakis, N. Lydatakis, S. Kateros, S. Georgiou, and P. Zikas. "Transforming Medical Education and Training with VR Using M.A.G.E.S." In: *SIGGRAPH Asia 2018 Posters*. SA '18. New York, NY, USA: Association for Computing Machinery, 2018. ISBN: 9781450360630. DOI: 10.1145/3283289.3283291.

[32] V. Pereira, T. Matos, R. Rodrigues, R. Nóbrega, and J. Jacob. "Extended Reality Framework for Remote Collaborative Interactions in Virtual Environments." In: *2019 International Conference on Graphics and Interaction (ICGI)*. 2019, pp. 17–24. DOI: 10.1109/ICGI47575.2019.8955025.

[33] T. Piumsomboon, G. A. Lee, J. D. Hart, B. Ens, R. W. Lindeman, B. H. Thomas, and M. Billinghurst. "Mini-Me." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. New York, New York, USA: ACM Press, 2018, pp. 1–13. ISBN: 9781450356206. DOI: 10.1145/3173574.3173620.

[34] G. D. Ruxton. "The unequal variance t-test is an underused alternative to Student's t-test and the Mann–Whitney U test." In: *Behavioral Ecology* 17.4 (May 2006), pp. 688–690. ISSN: 1045-2249. DOI: 10.1093/beheco/ark016.

[35] B. Sarupuri. "Comfortable and Efficient Travel Techniques in VR." In: *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings*. UIST '18 Adjunct. New York, NY, USA: Association for Computing Machinery, 2018, pp. 232–235. ISBN: 9781450359498. DOI: 10.1145/3266037.3266126.

[36] B. Sarupuri, S. Hoermann, F. Steinicke, and R. W. Lindeman. "Triggerwalking: A Biomechanically-Inspired Locomotion User Interface for Efficient Realistic Virtual Walking." In: *Proceedings of the 5th Symposium on Spatial User Interaction*. SUI '17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 138–147. ISBN: 9781450354868. DOI: 10.1145/3131277.3132177.

[37] T. Schubert. "The sense of presence in virtual environments: A three-component scale measuring spatial presence, involvement, and realness." In: *Zeitschrift für Medienpsychologie* 15 (Apr. 2003), pp. 69–71. DOI: 10.1026//1617-6383.15.2.69.

[38] Y. Sermet and I. Demir. "An Immersive Decision Support System for Disaster Response." In: *26th ACM Symposium on Virtual Reality Software and Technology*. VRST '20. Virtual Event, Canada: Association for Computing Machinery, 2020. ISBN: 9781450376198. DOI: 10.1145/3385956.3422087.

[39] H. J. Smith and M. Neff. "Communication behavior in Embodied virtual reality." In: *Conference on Human Factors in Computing Systems - Proceedings*. Vol. 2018-April. Association for Computing Machinery, 2018. ISBN: 9781450356206. DOI: 10.1145/3173574.3173863.

[40] M. Sousa, D. Mendes, and J. Jorge. "Safe Walking in VR." In: *The 17th International Conference on Virtual-Reality Continuum and Its Applications in Industry*. VRCAI '19. New York, NY, USA: Association for Computing Machinery, 2019. ISBN: 9781450370028. DOI: 10.1145/3359997.3365737.

[41] I. E. Sutherland. "A Head-Mounted Three Dimensional Display." In: *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*. AFIPS '68 (Fall, part I). New York, NY, USA: Association for Computing Machinery, 1968, pp. 757–764. ISBN: 9781450378994. DOI: 10.1145/1476589.1476686.

[42] M. Uhr, S. Haselmann, L. Steep, C. Bretschneider, and J. Eickhoff. "VRoadworks - Interactive Data Visualization for Coordinating Construction Sites in Virtual Reality." In: *Proceedings of Mensch Und Computer 2019*. MuC'19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 843–847. ISBN: 9781450371988. DOI: 10.1145/3340764.3345364.

[43] P. Vichitvejpaisal, N. Porwongsawang, and P. Ingpochai. "Relive History: VR Time Travel at the World Heritage Site." In: *The 17th International Conference on Virtual-Reality Continuum and Its Applications in Industry*. VRCAI '19. New York, NY, USA: Association for Computing Machinery, 2019. ISBN: 9781450370028. DOI: 10.1145/3359997.3365733.

[44] M. Zachara and J. P. Zagal. "Challenges for Success in Stereo Gaming: A Virtual Boy Case Study." In: *Proceedings of the International Conference on Advances in Computer Enterntainment Technology*. ACE '09. New York, NY, USA: Association for Computing Machinery, 2009, pp. 99–106. ISBN: 9781605588643. DOI: 10.1145/1690388.1690406.

Appendix

A

Characterization Questionnaire

The characterization questionnaire aims to give some insight into the technological literacy of the test subject and the test population in general.

Age The user's age.

Gender Multiple choice question with *Male*, *Female*, *Non-Binary*, *Prefer not to Answer* or *Other: Please fill in* as options.

Height The user's height.

Completed Education Multiple choice question with *Basic Education*, *Secondary Education*, *Higher Education - Bachelor's Degree*, *Higher Education - Master's Degree* or *Higher Education - Philosophy Degree* as options.

Dominant Hand Either *Left* or *Right*.

VR Experience Multiple choice question with *Never used it*, *Used sometimes*, *Uses every month*, *Uses every week* or *Uses everyday* as options.

Videogame Experience Multiple choice question with *Never played*, *Plays sometimes*, *Plays every month*, *Plays every week* or *Plays everyday* as options.

If you have experience with videogames, which of the following affirmations applies? Multiple choice questions where the user can choose multiple answers from the options which are; *Plays 3D games*, *Plays 2D games*, *Plays on consoles*, *Plays on computers*, *Plays in VR*, *Plays in Augmented Reality (AR)*, *Plays in mobile devices (smartphones/tablets)*.

Experience with Emergency Management Systems *Yes* or *No* question.

Dificulty seeing on a daily basis, even when wearing glasses or contact lenses? *Yes* or *No* question.

Appendix B

Consent Form for Testing Session

The testing session required participants to read and consent to participate in the session. Below is the consent presented to participants before starting the session.

"In order to participate in this testing session you must be aware of what will happen, what data will be gather and how it will be processed. You may only participate in this session if, after reading this information, you concent to participate in this study.

Health Considerations During the testing session you will have to wear a virtual reality headset. Since this headset will be worn by multiple participants, several hygienic precautions are taken: between each session the disposable face covering is replaced, the headset and controllers are disinfected with alcohol and the room is aired. And these sessions are conducted 24 hours apart. During the Covid-19 pandemic, these procedures are absolutely necessary. Wearing a virtual reality headset might cause the user to feel light-headed, nauseous, headaches, neck pain among other things. If any of these symptoms manifest, please inform the researcher so the session can be paused or terminated. You might also pause or terminate the session for any reason at any time, just close your eyes and ask to end the session.

Data Considerations All collected data will only be used in the context of this thesis. Data will be anonymized before being processed. The information collected will be the one present in the questionnaires, along with researcher observations, verbal communication during the session and data collected by the system (participant's position, rotation and execution times).

At any point during the session you might ask questions.

By choosing “Accept” below, you accept to voluntarily participate in this academic study, state that you read and understood the information hereby presented and accept that your data will be collected and processed as well as the health risks associated with participating.

Thank you.”

Appendix C

Explicit Hyperlinks

This appendix contains a list of every hyperlinked referenced across this document written explicitly, in alphabetic order.

- ACM Symposium on Virtual Reality Software and Technology
  https://vrst.acm.org/
- AltspaceVR
  https://altvr.com
- Ambulance By pictohaven
  https://thenounproject.com/icon/882729/
- Cardboard Design Lab
  https://play.google.com/store/apps/details?id=com.google.vr.cardboard.apps.designlab
- Centro de Informação Geoespacial do Exército
  https://www.igeoe.pt
- CNET: Oculus Quest reivew
  https://www.cnet.com/reviews/oculus-quest-review/
- Designing for Google Cardboard
  https://designguidelines.withgoogle.com/cardboard/
- DOTween by Daniele Giardini
  http://dotween.demigiant.com
- GDAL
  https://gdal.org

- High Fidelity
  https://www.highfidelity.com
- Human Interface Guidelines
  https://developer.apple.com/design/human-interface-guidelines/
- IEEE Conference on Virtual Reality
  http://ieeevr.org
- JSON .NET For Unity by ParentElement
  https://www.parentelement.com/assets/
- Kenney's Car Kit
  https://www.kenney.nl/assets/car-kit
- Kenney's Fantasy Town Kit
  https://www.kenney.nl/assets/fantasy-town-kit
- Krita
  https://krita.org/
- Laser Pointer By Robin Wilde
  https://thenounproject.com/icon/2836402/
- Lighthouse tracking examined
  http://doc-ok.org/?p=1478
- Measuring Usability With The System Usability Scale (SUS)
  https://userfocus.co.uk/articles/measuring-usability-with-the-SUS.html
- Mirror Networking
  https://mirror-networking.com
- Mirror Networking - Remote Actions Documentation
  https://mirror-networking.gitbook.io/docs/guides/communications/remote-actions
- NaughtyAttributes
  https://github.com/dbrizov/NaughtyAttributes
- Noun Project
  https://thenounproject.com
- Nvidia Holodeck
  https://www.nvidia.com/en-us/design-visualization/technologies/holodeck/
- Open Source Routing Machine
  https://github.com/Project-OSRM/osrm-backend

- Open Street Map
  https://www.openstreetmap.org/
- OptiTrack
  https://www.optitrack.com
- Paint Brush By shuai tawf
  https://thenounproject.com/icon/3029766/
- Photon PUN
  https://www.photonengine.com/en-US/PUN
- QGis
  https://www.qgis.org/
- Ruler By André Luiz Gollo, BR
  https://thenounproject.com/icon/332184/
- Shapes by Freya Holmér
  https://acegikmo.com/shapes/
- Snaps Prototype - Office
  https://assetstore.unity.com/packages/3d/environments/snaps-prototype-office-137490
- syGlass
  https://www.syglass.io
- Texture Painting by Shahriar Shahrabi
  https://shahriyarshahrabi.medium.com/mesh-texture-painting-in-unity-using-shaders-8eb7fc31221c
- The Lion King: Reinventing the Future of Virtual Production.
  https://resources.nvidia.com/gtcd-2020/GTC2020-s22376
- Unity Terrain System Documentation
  https://docs.unity3d.com/2019.4/Documentation/Manual/terrain-Heightmaps.html
- Virtual Reality Toolkit
  https://www.vrtk.io
- VRCollabCrisis - Gitlab
  https://gitlab.com/vrcollabcrisis/
- VR Design Best Practices
  https://developer.oculus.com/design/bp-generalux/
- VR-1
  https://segaretro.org/VR-1

- Vrui VR
  https://arsandbox.ucdavis.edu
- Wallace & Gromit: The Big Fix Up - Recruitment Drive
  https://www.youtube.com/watch?v=auRcq7gI0nI
- Wind By Bernar Novalyi
  https://thenounproject.com/icon/1671599/

ANNEX I

System Usability Scale (SUS) Questions

Likert-scale questions from the SUS questionnaire developed by John Brook [5], the answer to which question is a number between one and five, where one meaning *Strongly Disagree* and *Strongly Agree*, respectively.

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

ANNEX II

Igroup Presence Questionnaire Questions

Likert-scale questions from the *Igroup Presence Questionnaire* developed by Thomas Schubert [37].

1. In the computer generated world I had a sense of "being there"
2. Somehow I felt that the virtual world surrounded me.
3. I felt like I was just perceiving pictures.
4. I did not feel present in the virtual space.
5. I had a sense of acting in the virtual space, rather than operating something from outside.
6. I felt present in the virtual space.
7. How aware were you of the real world surrounding while navigating in the virtual world? (i.e. sounds, room temperature, other people, etc.)?
8. I was not aware of my real environment.
9. I still paid attention to the real environment
10. I was completely captivated by the virtual world.
11. How real did the virtual world seem to you (Compared to reality)?
12. How much did your experience in the virtual environment seem consistent with your real world experience?
13. How real did the virtual world seem to you (Compared to imagination)?
14. The virtual world seemed more realistic than the real world.

Annex III

Supporting Technologies

This annex lists other tools, assets and resources that helped the development of this thesis but aren't directly related to virtual reality, networking or geographic information systems.

- DOTween by Daniele Giardini Animation engine used for animations throughout the application. - Last accessed: 04/05/2021
- JSON .NET For Unity by ParentElement JSON serialization library used to build the importer for GeoJSON data. - Last accessed: 04/05/2021
- Kenney's Car Kit 3D assets used to build 3D vehicle markers. - Last accessed: 04/05/2021
- Kenney's Fantasy Town Kit 3D assets used to build town markers. - Last accessed: 04/05/2021
- NaughtyAttributes Unity Inspector extension used throughout development to improve asset inspectors. - Last accessed: 04/05/2021
- Noun Project Royalty free icons, used for toolbox icons. - Last accessed: 04/05/2021
  - Wind By Bernar Novalyi - Last accessed: 04/05/2021
  - Ambulance By pictohaven - Last accessed: 04/05/2021
  - Ruler By André Luiz Gollo, BR - Last accessed: 04/05/2021
  - Paint Brush By shuai tawf - Last accessed: 04/05/2021
  - Laser Pointer By Robin Wilde - Last accessed: 04/05/2021
- Shapes by Freya Holmér A real-time vector graphics library, used throughout the project, from layers to markers to notification indicators. - Last accessed: 04/05/2021

- Snaps Prototype - Office Themed assets used to build the virtual environment. - Last accessed: 04/05/2021
- Texture Painting by Shahriar Shahrabi Blog post explaining how texture painting could be implemented, used as inspiration for the *Brush Tool*. - Last accessed: 04/05/2021