POLITECNICO DI TORINO

COLLEGIO DI INGEGNERIA INFORMATICA, DEL CINEMA E MECCATRONICA

Master Degree in Computer Engineering

Master Degree Thesis

Immersive Analysis of Hierarchical Semantic Datasets

Supervisors
Prof. Valentina Gatteschi
Prof. Fabrizio Lamberti

Candidate
Salvatore Calogero Di Cara
matricola: 251608

December 2019

. . . .

Salvatore Calogero Di Cara
Turin, October 10, 2019

Sommario

I recenti avanzamenti tecnologici nell'ambito della *Realtà Virtuale* (RV) hanno reso degli innovativi ed efficienti visori sufficientemente economici da poter essere alla portata di un ampio pubblico. Oltre alle classiche applicazioni in RV, dedicate soprattutto all'intrattenimento e alla simulazione, gli studi in un nuovo campo di ricerca, detto *Immersive Analytics*, hanno evidenziato come le moderne tecnologie di visualizzazione e interazione abbiano un grande potenziale per creare degli utili strumenti di analisi dei dati. In quest'epoca, in cui la quantità di dati disponibili continua a crescere esponenzialmente, può essere di fondamentale importanza riuscire a estrarre informazioni utili da grandi collezioni di documenti. Inoltre, un'applicazione che rappresenti queste informazioni in maniera stimolante potrebbe suscitare l'interesse non solo degli esperti di dominio e degli analisti, ma anche degli utenti generici. Il presente lavoro di tesi si occupa di discutere la progettazione e lo sviluppo di una piattaforma che, sfruttando le moderne tecnologie di RV, fornisca un ambiente immersivo per la visualizzazione, l'analisi e la condivisione di dati relativi a dei documenti. In particolare, l'informazione è stata rappresentata mediante termini raggruppati in categorie semantiche. L'obiettivo è di fornire agli utenti un utile strumento che permetta loro di accrescere la conoscenza relativa a un certo tema e di individuarne facilmente gli argomenti principali. Le caratteristiche fondamentali dell'applicazione realizzata includono un'accessibilità distribuita basata sul web, portabilità su diversi dispositivi, interazioni naturali con l'ambiente virtuale, meccanismi di collaborazione asincrona e menu che permettono di modificare la categorizzazione rappresentata. Il progetto è stato condotto in collaborazione con il gruppo di ricerca DEI dell'Università *Carlos III* di Madrid. I membri del gruppo, in virtù della loro esperienza accumulata, hanno supervisionato il lavoro e fornito i requisiti necessari alla sua realizzazione. Inoltre, sono stati condotti dei test di valutazione della piattaforma, stimandone l'usabilità e mettendo in luce i vantaggi derivanti dall'utilizzarla in un ambiente immersivo, come quello offerto da un visore per la RV. I risultati di tali test verranno discussi nella parte finale di questo documento.

Abstract

The recent improvements in *Virtual Reality* (VR) technologies made novel and efficient headsets cheap enough to be affordable for the wide public. In addition to the classic VR applications, mainly with entertainment and simulation purposes, studies in the emerging research field of *Immersive Analytics* have highlighted the great potential of modern visualization and interaction techniques to create engaging data analysis tools. In this era, where the amount of available data keeps growing exponentially, it can be crucial to extract useful knowledge from large collections of documents. Furthermore, an application to depict such information in a stimulating way can arouse enthusiasm not only in domain experts and data analysts but also in casual users. The present thesis work discusses the design and development of a platform which, exploiting modern VR technologies, provides an immersive environment for the visualization, analysis, and sharing of document-related data. In particular, information has been represented as terms grouped into semantic categories. The goal is to provide users with an useful instrument that allows them to increase their knowledge on a certain subject and easily detect main topics. The key features of the realized application include a distributed web-based accessibility, portability on several devices, natural interactions with the virtual elements, asynchronous collaboration mechanisms, and menu interfaces to edit the presented categorization. The project has been conducted in collaboration with the DEI research team at University *Carlos III* of Madrid. The members of the team, thanks to their previous experience, supervised the work and supplied the requirements for its realization. An evaluation of the platform has also been conducted, assessing its usability and pointing out the benefits brought by using it in an immersive environment, such as the one offered by a VR headset. The results of this evaluation will be discussed in the final part of the present document.

Acknowledgements

I would like to acknowledge Prof. Paloma Díaz and PhD. Teresa Onorati from University Carlos III of Madrid, who supported the development of this research.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This first chapter aims to outline the fundamental concepts on which this thesis is based. At first, the main notions concerning Virtual Reality will be supplied, along with an overview of the technologies implementing it. Then focus will move on the emerging research field that inspired this work, later called Immersive Analytics. The motivations and goals that led this research will be described hereafter. Finally, the structure of the present document will be summarized.

1.1 Virtual Reality

Virtual Reality (VR) is a simulated experience in which a subject is placed in an environment similar to or significantly different from the real world[1]. The term ”virtual” has to be intended as ”not physically existing but made to appear by software”[2]. Such technology comes with two form factors: wearable headsets or multi-projected room-sized cubes. VR exploits realistic images, audios, videos, and haptic feedback in several ways to stimulate users’ senses and trick them into the realism of the situation. Moreover, users can typically move within the generated environment, look around, and interact with fictional items.

[1]https://en.wikipedia.org/wiki/Virtual_reality

[2]https://www.etymonline.com/search?q=virtual

The boundary line among VR and the real world, though, is not so well-defined. To be specific, Milgram et al. [1] described a more complex reality-virtuality continuum, defining a continuous scale that ranges from the completely real to the completely virtual. Everything in between these two ends is identified as *Mixed Reality* (MR). MR includes the technologies of *Augmented Reality* (AR), in which virtual elements are used to augment the real, and of *Augmented Virtuality* (AV), where likewise the real is used to augment the virtual.
There are many applications for VR technologies. They vary from entertainment media (e.g. games, movies) to educational purposes (e.g. military, medical and spatial simulations).
However, VR still suffers from several issues. When people are exposed to it for too long, they can lose spatial awareness of their surroundings and bump into real-world objects [2]. Furthermore, VR headsets may cause a reduction of the lacrimation, with consequent eye fatigue [3]. Finally, a disconnection between what the users see and what their body perceives can result in motion sickness, along with many symptoms as general discomfort, nausea, headache, dizziness, and disorientation [4].

1.1.1 Immersion and presence

According to Slater et al. [5], two concepts are of the utmost importance to quantify the quality of a VR experience: *immersion* and *presence.*
Immersion is tightly related to the used VR technology and concerns objective characteristics. It represents a technology capability to provide a realistic experience, making users temporarily unaware that they are inside a simulation. Immersion can be described through six attributes:

- *Inclusion*: it measures how much the physical reality is left outside the simulation.
- *Extension*: the grade of multi-sensory involvement (e.g. haptic support, sound effects).
- *Surrounding*: the extension provided by the field of view.
- *Vividness*: the accuracy of the displayed information (e.g. display resolution, variety of colors).

- *Match*: the correlation between the information displayed in VR and the user's perceived feedback on the body.
- *Plot-interactivity*: it measures how much an user can influence the events in the virtual environment.

Presence, on the other hand, is related to the human user. It measures the sense of being in the virtual space, even though one is not physically there. Presence is more subjective and qualitative, describing the psychological impact of living a virtual experience. This sort of illusion is carried out by acting on four senses:

- *Sense of believing*: it takes into account how much users feel to be in a remote place, rather than just looking at images. It can be obtained through accurate 3D models, good device calibration, high frame-rate, and low latency.
- *Sense of naturalism*: how realistic an interaction is perceived by users. Force feedback generated from controllers can give the illusion of the physical nature of the environment.
- *Sense of social presence*: the feeling of being in a virtual body. It does not require to use an extremely accurate 3D model for the avatar, but its movements have to be consistent with the user's ones.
- *Sense of co-presence*: how much users perceive themselves together with other virtual avatars, both tied to other users or computer-generated.

1.1.2 VR technologies overview

As mentioned, VR displays mainly come in two categories.
The first category is represented by CAVE-like systems, from the name of the first project that realized such a tool [6]. The acronym is recursive and stands for "Cave Automatic Virtual Environment". As can be seen in Figure 1.1[3], these systems provide a room-sized visualization apparatus. Such a kind of technology consists of a set of fixed wall screens, along with computer-driven projectors, and stereoscopic polarized LCD glasses. The user's position

[3] Image by Davepape - Own work, , https://commons.wikimedia.org/wiki/File:CAVE_Crayoland.jpg

is tracked through infrared cameras. According to that, the computers generate two images, one for each eye, and project them on the wall screens so that the 3D effect can be experienced through glasses. A tracked controller, called *wand*, can also be used to interact with the environment.

Figure 1.1: An user experiencing VR within a CAVE environment

The second category consists of head-mounted displays[4] (HMD), that are helmet with small binocular, or monocular, LCD, or OLED, screens placed directly in front of the user's eyes. Such stereoscopic displays show computer-generated imagery (CGI), enabling the VR experience. Moreover, in the so-called see-through HMDs, CGI can be projected on a partially reflective mirror, allowing also MR experiences. Head movements are tracked through optical sensors, and the displayed CGI is consequently updated, allowing users to look around. Such devices may simply track rotational movements, enabling 3-degrees-of-freedom (3DOF) or, in a more sophisticated way, they can track both position and rotation, enabling 6-degrees-of-freedom (6DOF). Hand-tracked controllers can also be exploited to naturally interact with content.

For a long time, due to the better resolution, higher image quality, wider field of view, lower latency, see-through glasses, a more collaborative environment,

[4]https://en.wikipedia.org/wiki/Head-mounted_display

and lightweight devices, CAVE-like systems have been widely preferred over HMDs by most researchers[5]. Their cost, though, is prohibitive and the privilege of having such a system is restricted to few labs. Furthermore, while a few years ago HMDs, input devices, and tracking sensors were sold separately or were custom-made, nowadays fully integrated systems, such as *Oculus Rift*, *HTC Vive*, and *Google Daydream*, have become relatively cheap commercial products, simpler to set up and compatible with average-graphics computers. Furthermore, devices like *Google Cardboard* and *Samsung GearVR* allow to turn even a generic smartphone into an HMD. Plus, modern HMD increasing performances have also been filling the technological gap with CAVE-like systems [7]. Also, many tools to develop and deploy cross-platform VR applications, such as *Unity3D*, *SteamVR*, and *WebVR* have been released, fostering a renewed interest for these technologies.

1.2 Immersive Analytics

In the following, the chronological events that led to the establishment of this new area of research will be discussed. Then, a proper definition of Immersive Analytics and its actual goals will be provided.

1.2.1 Historical Context

Several researchers have been concerned about the efficiency and simplification of interaction between humans and machines ever since the universal potential of computers began to emerge. Already in the late 1960s, Ivan Sutherland presented at Harvard the first head-mounted three-dimensional display system [8], a prototype of mixed-reality, and a light pen interacting with a graphical display [9].
The so-called field of *Human-Computer Interaction* (HCI) has then developed, involving knowledge from previously unconnected disciplines such as computer science, cognitive perception, design, ergonomics, and human-factors engineering.
Researches on HCI are closely related to hardware innovation, bringing to an evolution of interaction devices.

[5]An example of comparison of the latest CAVE and HMD technologies can be found at: http://www.visbox.com/technology/cave-vs-hmd/

Alongside with input devices, also display technologies played a crucial role in improving interaction techniques. Despite Sutherland's prototypes, in fact, only by the 1990s, when the advances in graphics hardware and real-time rendering became significant, the first interactive virtual reality systems (e.g. CAVE [6]) were built. Since then, interest in interacting with virtual content has been growing. Several conferences on VR and 3D User Interfaces indeed have been progressively held, starting from the early *IEEE Virtual Reality* (IEEE VR), established in 1993, to the more recent *IEEE Symposium on 3D User Interfaces* (3DUI) (2006) and *ACM Symposium on Spatial interaction* (SUI) (2013). In 2018, the first two merged into the *IEEE Conference on Virtual Reality and 3D User Interfaces*.
The studies on visualization applied to scientific research were initially focused on data that had embedded spatial features, like ocean currents, architectural reconstructions, blood-density scans, and fluid flow. Nevertheless, in 1991, Card et al. [10] began experimenting with the immersive visualization of more abstract data, e.g. merely numerical, representing them in a spatial dimension.
Analogous researches upon graphics [11], statistics [12, 13] and HCI [14], established the basis for a new sub-field of visualization, later called Information Visualization (*InfoVis*), aiming to represent and explore abstract data by the means of computer graphics. The *IEEE Symposium on Information Visualization* (IEEE InfoVis) has been instituted in 1995 to focus on this topic, and in 2007 it further became a conference (IEEE VIS), leading to many publications throughout the years [15].
Even though *InfoVis* researchers were initially interested in immersive 3D, since those technologies were only available in few labs, they then started to focus on 2D desktop representations. This trend was supported also by some studies finding 3D less useful than 2D for abstract data visualization [16] [17].
In the 2000s, the quick growth in data volumes started to require more complex analysis techniques, together with more useful visualizations. Thomas and Cook [18] theorized, through 19 guidelines, an analytical methodology supported by interactive visual interfaces and called it *Visual Analytics* (VA). As has been pointed out, new visualization techniques should be able to scale to huge amounts of data and to enable high-level analysis, rather than focusing on low-level tasks, exploiting interactive visual interfaces. However, this definition of VA was not constrained to any specific technology. VA researchers, though, were still skeptical about virtual environments and kept

developing for desktop environments.
Researchers only recently began reconsidering the latest interaction and display technologies to be adequate to address the VA agenda, stimulating our senses and representing a valid alternative to mouse and keyboard. At IEEE VIS 2014, a workshop, provocatively titled "*Death of the Desktop: Envisioning Visualization without Desktop Computing*", has been held on this topic[6].
Consequently, the term "*Immersive Analytics*" (IA) has been created and introduced in 2015 at the *IEEE Symposium on Big Data Visual Analytics* (IEEE BDVA) by those who were investigating data visualizations, using multi-sensory interactions, in virtual and mixed reality [19].
After that, similar workshops were held at Shonan, Japan [20] and Dagstuhl, Germany [21], to discuss intensively on this topic. Since then, interest in these opportunities has been growing quickly, many other workshops were held, and at IEEE BDVA 2018, extended for the occasion to *Big Data Visual and Immersive Analytics*[7], the community was explicitly invited to develop and submit original IA applications.

1.2.2 What is Immersive Analytics?

Immersive Analytics is a research field that explores how innovative interaction and display technologies can bring support to data analysis and decision making, rendering them accessible to everyone, individually or collaboratively, and everywhere [19]. One of its purposes, indeed, is to extend data analysis tools, that have always been a prerogative of scientists, policymakers, and business analysts, to a wider general public. Various applications have already been realized to support technology, marketing, science, healthcare, emergency management, and other domains.
Tricks and guidelines used for game and UI design can be exploited to place users in immersive environments, within their data, allowing them to live the analysis process as an exciting experience.
This field is not tied to any particular technology. However, it has been mainly fueled by the various improvements made by VR and AR technologies in recent years, along with the advancements in sensors and machine learning techniques, that allowed more detailed ways to detect speech and gestures.

---

[6]http://dataphys.org/workshops/vis14/tag/visualization/

[7]http://bdva.net/2018/

According to Dwyer et al. [22], IA offers many opportunities compared to more traditional visual analytics, including:

- *Embodied data exploration*: abandoning mouse and keyboard for touch, gestures, and voice, the data exploration should become more engaging and intuitive.
- *Collaboration* between users: either local or remote, synchronous or asynchronous, it can get deeper, more interesting and equitable.
- *Spatial immersion*: since there is no more desktop, users can place their objects in a three-dimensional workspace, allowing both 2D and 3D visualizations.
- *Multi sensory presentation*: beyond the sight, audio and the other senses (e.g. haptic feedback) can be used either to provide further information or as an alternative to vision.
- Immersive scenarios can lead to an increased *engagement* and *commitment* from general public on sensitive subjects (e.g. climate change).

1.3 Motivations and goals

In the Big Data era, it is crucial to build ad-hoc tools to extract useful knowledge from constantly growing unstructured collections of news, articles and social network messages [23]. For this purpose, clustering algorithms are frequently employed to detect and categorize important words from a corpus of documents. The result of this process is commonly a hierarchical semantic model. Furthermore, graphical and interactive representations of data have proven to bring several advantages to collaboration, decision making, and information sharing [24]. *Visual Analytics* guidelines, though, should be followed to make such visualizations useful for supporting human users in gaining precious insights from large datasets [18].
According to Shneiderman [25], the process of seeking for visual information can be summarized by the mantra *"overview first, zoom and filter, then details on-demand"*. Therefore, when it comes to big and complex amounts of data, even basic actions, such as selecting and filtering, can become hard to perform in traditional 2D environments. That is why an IA approach has been adopted and the present thesis stands as a contribution in this field. This work

is indeed part of a broader research in the field of IA, carried out by the *Laboratorio De Sistemas Interactivos* (DEI) research group at University *Carlos III* of Madrid.
More specifically, this project aims to develop a platform which, exploiting modern VR technologies, provides an immersive environment for the visualization, analysis, and sharing of semantic data clusters, to identify the important topics and to increase the general knowledge about a specific domain. The main focus will be on the benefits brought by such a kind of visualization to users' comfort, engagement, satisfaction, collaborative working and decision making.
Starting from these premises, an immersive bubble chart has been realized, improving the tool described by Díaz et al. [47]. In this model the information is represented by the means of semantic bubbles, abstracting terms and categories. These bubbles directly surround users and several tasks can be executed on each of them to extract information and to find relationships among data.
The proposed enhancements include increased portability of the application, mechanisms to support collaboration, novel methods to interact with the environment, faster navigation techniques, and additional interfaces to correct the underlying taxonomy. An evaluation of users' performance and experience using this application has also been conducted and results will be further discussed.

1.4 Chapters overview

The present document is divided into seven chapters. Besides this general introduction, they all aim to outline different phases of the project that has been carried out.
Chapter 2 describes a preliminary research stage on the early provided solutions concerning the Immersive Analytics field, including the previous researches led by the DEI team on the same topic. This was necessary to outline the background of this proposal, to overcome prior limitations, and to propose novel contributions.
In Chapter 3 are presented the main requirements that have determined specific design choices and the selection of adequate technologies. These requirements have been delivered also by some members of the DEI team, taking into account their previous exploratory studies and the current state of the art.

The software and hardware technologies employed to build the proposed analysis tool are discussed in detail within Chapter 4.
Chapter 5 focuses on the whole development process, explaining the operation of the implemented application and providing details on its different functionalities.
Furthermore, in Chapter 6 are shown the results of some experimental tests conducted on the application, comparing two different setups. Their purpose was to assess the software usability and the impact of several features on users' performances.
The thesis ends with Chapter 7, containing the conclusions and some perspectives for future work.

Chapter 2

State of the Art

In this chapter, an analysis of the previously realized solutions in the field of *Immersive Analytics* (IA) will be conducted, to outline the context in which this thesis fits.
Then, the focus will move on the works that inspired the current project and established the basis for future improvements.
Finally, the different issues and approaches presented will be discussed, to detect which are the possible choices involved and the novel features introduced by the present thesis.

2.1 Literature overview

IA is an interdisciplinary sector of knowledge, bringing together experts from visual and data analytics, human-to-computer interaction, virtual and mixed reality.
Enthusiasm for this topic has been growing only in recent years, though. Until now, 3D interface researchers have been focusing more on low-level issues, like body tracking and gesture recognition [19]. A smaller number of studies indeed has been made, in different scenarios of application, on the benefits brought by VR immersive visualizations to data analysis and scientific research. The support they brought to collaboration has been briefly discussed as well [31].
However, there are no universal guidelines for abstract data representations in immersive environments yet, since the design is tightly related to the type of data, the tasks to perform, the device technology and the dimensions of the display.

Here we analyze the various IA solutions found in literature, pointing out the advantages and limitations that inspired the design choices of this work.

2.1.1 A virtual reality visualization tool for neuron tracing

In this study, Usher et al. [26] developed, in collaboration with trained neuroanatomists, a VR system which allows to directly interact with neurons within visual cortex microscope scans, editing the branches of a trace.
The tool was designed to support neuroscientists in tracing neurons with less physical and mental fatigue than with desktop applications. VR environment also allowed neuron tracing in datasets larger than the ones supported by standard approaches. An *HTC Vive* System has been used in order to support room-scale (i.e. users can walk around or into the data) and to have wand-style controllers, useful to naturally and intuitively draw neuron traces. Apart from drawing traces, other interesting features were the possibility to record replays of the users' sessions and a mini-map which helped users not to get lost during navigation.
Analyzing the specialists' accuracy and speed in detecting and correcting the connectivity of neurons, an evaluation between the tool and a similar state-of-the-art 2D application, *NeuroLucida*, has been conducted. Based on their experience with the 2D tool, the people who attended this test were divided among novice and experts.
User performances were found to be acceptably the same in both scenario. Novice users performed 2x faster in VR than 2D, while only in few and particular cases experts performed better in *NeuroLucida.*
Results showed that users generally found the VR system more intuitive, allowing them to focus more on the data and feeling less fatigued. Even if the ability of moving around was given, most users showed to prefer grabbing items and bringing them closer, commenting that they felt more productive while seated. Therefore, experts appreciated analyzing the data volume from different perspective. Mini-map was found somewhat useful by users, but they actually looked at it very rarely during tracing. Expert users also suggested introducing original 2D microscope images within the data volume, since they anyway felt familiar with those representations. Both novice and experts users strongly suggested the option of re-editing previous sessions. System menus shall be added to address this purpose, but this may overload users' cognitive

load, possibly reducing productivity.

2.1.2 Visualization of vector field by virtual reality

This study, developed in 1999 by Kageyama et al. [27], aimed to build a vector field in a virtual environment, representing one of the early works in immersive visualization.
Due to the historical context, the technology used was a CAVE-like, projector based, room-sized VR system, called *CompleXcope*, customized by University of Illinois, Chicago. Users in the *CompleXcope* room had to wear stereo glasses and to hold a 3D mouse called *wand*.
Even if the research is now pretty old and our focus is not on vector fields, an interesting feature of this application was the possibility to physically walk around data. Indeed virtual environment was projected on room walls, giving a true sense of immersion.
An other interesting and clever characteristic was a visual menu with several options to manipulate vectors. When one pressed a button on the wand, a menu appeared in front of the user, some meters away. Options could be selected by shooting them with a virtual laser beam emitted from the wand. Menu could be hidden on-demand as well.
Therefore, some limitations of this solution were the exclusivity, customization and high-cost of the VR apparatus. Unfortunately, no evaluation for the application was provided.

2.1.3 An immersive visualization study on molecules manipulation

In this research, led by Da Costa and Nedel [28], it was studied the impact of an immersive visualization in manipulating and interacting with molecules. Inspired by the current challenges addressed by Immersive Analytics (e.g. simpler interactions, no screen limitations, increased perception) and by other state-of-the-art applications for molecular visualization (e.g. *VMD*, *PyMOL*, *UnityMol*), researchers developed an alternative solution using *Unity3D*, *Meshlab* (to handle protein models before they could be exported to Unity), *Oculus Rift* and *Razer Hydra* controllers. They meant to study the benefits of an immersive environment in terms of increased *speed* in executing tasks, *fewer errors*, due to less distractions, and more *comfort* from users.
In order to evaluate the work, two setups were build: one for the immersive

approach and other for the non-immersive one. An Head-Mounted-Display (HMD) and two 3D Interface Controls were used for the first setup. An LCD monitor, two 3D Interface Controls and a mouse (to look around) were used in the second setup. In both cases user is seated on a caster free-rotating chair, indeed HMD can cause less sickness than standing, especially if users need to move and rotate.
Atoms were modeled as spheres, aggregated into molecular structures. Interactions with molecules were designed to be as natural, intuitive and close to human movements as possible. Using the analog sticks users could move around freely in the virtual environment. Moreover, several actions could be performed on each atom: *selection*, pointing a laser beam emitted from controllers, *grab*, pushing/releasing the trigger button,*toggle visibility*, *rotate*, *increase/decrease size* and *undo all changes*, restoring the original molecular structure.
Evaluation involved 6 men and 5 women with background on chemistry or biotechnology and different experience with the alternative products previously mentioned. Three kinds of tests were conducted using both setups. In the first one, with DNA, users had to move several atoms, in order to evaluate the speed at which them can select, grab and move. In the second one, with GFP, users had to locate some specific atoms, in order to study attention and precision. In the third one, users had to grab and move around atoms to locate and show only the iron atoms of the hemoglobin structure, in order to evaluate similar functions offered by *UnityMol*. Questionnaires were used to evaluate comfort.
Results showed that users made fewer mistakes and had a smaller execution time when using the immersive approach. Not so many users rated as high-comfortable the non-immersive experience, while an higher percentage of them declared to feel very comfortable in the immersive environment. An interesting finding was that in the non-immersive setup mouse was never used to move around and participants always preferred rotating the whole molecular structure. Thus, camera mostly remained stationary in that setting, whereas users rotated their head more in the immersive approach. Researchers claimed this to be due to a more natural interaction and more freedom in movement than with a regular display.
Some improvements were also recommended, through questionnaires, for the

interaction mechanics and the molecular representation. Authors also suggested that others sectors of knowledge could take advantage of the immersive visualization approach.

2.1.4 Immersive and collaborative data visualization using virtual reality platforms

In 2014 Donalek et al. [29] addressed the issue of multi-dimensional data representations. They highlighted the benefits brought by immersive virtual reality platforms in finding patterns and enhancing collaboration.
In a first section, current VR off-the-shelf technologies, such as *Second Life*, *OpenSim* and *vCaltech* were taken into account to build a solution. This approach presents some pros, since rendering, geometry and interaction issues are already solved and all is needed is application scripting. Several digital sky surveys were used as input dataset, since they can be represented as feature vectors in a multi-dimensional space. The features were then mapped to data objects having different graphical attributes, each one depicting a dimension: spatial coordinates, color, size, transparency, shape, orientation, texture, etc. The resulting application, realized in *vCaltech*, was able to visualize up to 10^5 data points. Objects in VR were also linked, through clickable hyperlinks, with external catalog information, which should support visual analysis. Flat screens were used to provide such additional data and they did not seem to compromise the quality of interaction. Researchers noted how collaboration is embedded by design in virtual environments.
It has been pointed out as off-the-shelf technologies needed specific browsers and were not optimized for rendering of huge datasets, so a second visualization tool, called *iVIZ*, was entirely realized using Unity3D. It was multi-platform and could also run in a web-browser. Indeed, it has been suggested how the use of a browser as an interface may encourage immersive VR among those who are reluctant to ”game-like” environments. The *iVIZ* platform supported all the functionalities provided by the first-described tool, plus a broadcasting function allowing scientists to share views on data and the capability to render up to 10^6 data points. Along with Leap Motion and Kinect, the environment could also be explored by the means of Oculus Rift goggles. However, when wearing HMD, since keyboard interface is precluded, no interaction mechanisms with virtual objects and data displays were available.

Additionally, authors provided a comparison between immersive visualization and 2D for a specific use case: reconstructing Martian landscape. Specialists' performances were evaluated on the basis of map drawings, made on an ad-hoc editor, realized after having examined image mosaics with both approaches. Researchers thought that more correct drawings had to be ascribed to increased spatial awareness. Scientists performed better in immersive environment, but curiously self-reported the two conditions as equal.

2.1.5 Immersive visualization of abstract information: an evaluation on dimensionally-reduced data scatterplots

This work, realized by Wagner et al. [30], extended the discussion on Immersive Analytics taking into account abstract data representations such as scatterplots.
These visualizations are usually applied on multidimensional data, along with dimensionality reduction (DR) methods (e.g. PCA), in order to generate a more compact version of the same information. Distance in a lowered-dimension space is crucial and 3D should allow to better differentiate depths between near points, leading to less perception errors than 2D. However, opinions on 3D in literature are mixed and challenges like navigation problems, perspective distortion and occlusion have brought many researchers to lose interest in it.
The purpose of this study was to compare, using a task-based empirical evaluation, scatterplot performances in 3 different setups: a desktop-based 2D visualization, a desktop-based 3D visualization and a HMD-based 3D immersive visualization (IM). All these three representations were realized using Unity. In the last two setups navigating the environment was also possible, using mouse and keyboard or joystick controllers. Ground, sky and an above light were included in the virtual environment for orientation purpose. Two different datasets were used for the analysis: one with more potential information gain in 3D and one without it. 30 subjects participated to the evaluation.
Results showed how users tend to move both hands together when interacting with points. Perception errors showed to be more or less the same in all three scenarios, contradicting one of the research hypothesis. Task completion time was 3 times slower in IM than in 2D. Researchers attributed this to the additional third-dimension to interact with. Users complained that navigation

speed in 3D was slower than in IM, however, no significant difference in execution time has been registered. IM was pointed out to be the most engaging setup, maybe due to the novelty of display and interaction technologies, or because of immersion and egocentric view actual effectiveness. Concerning tasks with similar performances in 3D and IM, users claimed that finding information in IM required less effort than 3D, due to a faster navigation, but this difference was not registered in their performances. Subjectively perceived accuracy also increased in IM, but this could also lead to over-confidence in observing errors. An interesting finding was that, as a consequence of navigation in IM, 40% of users suffered simulator sickness.
Authors also supposed that manipulating points from a fixed position would sacrifice the benefits of an egocentric point of view, but could minimize task completion time and reduce simulator sickness.

2.1.6 Building immersive data visualizations for the web

In this paper from late 2017, Butcher and Ritsos [32] investigated the importance for Immersive Analytics of the standard web technologies for VR. The term *Ubylitcs* has been coined to advocate that data analytics applications should be available anywhere at any time. It is stressed how the Web, indeed, fits the growing necessity for an ubiquitous multi-platform access to data. Deploying a VR data visualization tool on the Web does not require users to install any extra software, except for a browser, since the application can be reached through an URL. Furthermore, different libraries such as *D3.js*, which can easily bind arbitrary data to DOM elements, and *Three.js*, which allows to create VR visualization for the Web, are discussed. It is pointed out how the number of tools to support WebVR development is slightly increasing, recently leading to the creation of more complete frameworks like *A-frame* and *ReactVR*, announced by Facebook. WebVR enables portability to various devices, such as smartphone and PCs, in both HMD and mono-3D visualization modes.
The aim of this study was to provide a prototype for low cost immersive analytics, specifically a 3D bar chart, using web technologies. A first concept was built using WebVR polyfill (i.e. extra plugins for browser) and *Three.js*. Although its simplicity, it showed good performances, rendering 1000 polygons at 60 fps. Further investigations lead to a second prototype, built using *A-Frame* and *React.js* (a state-component framework to build UIs). It implemented new functionalities like gaze-based hovering and labeled bars. In both

prototypes design it has been underlined the importance of ambient light, shadows, background and rendering performance.
Unfortunately, no evaluation of the applications has been provided.

2.1.7 Immersive collaborative analysis of network connectivity: Cave-style or head-mounted display?

In 2017, Cordeil et al. [33] compared HMDs (e.g. *Oculus Rift*) and CAVE-like systems in terms of functionalities, user experience and increased collaboration. Considering that HMD are cheaper and potentially ubiquitous, the aim of their work was to investigate if the high cost of CAVE facilities was still justified. In particular, few research focused on the benefits brought by collaboration in immersive environments to abstract data representations.
Hence, a specific use case was presented: analysis of connectivity among network nodes. Two platforms were built in order to allow a fair comparison, along with an empirical evaluation model based on different tasks. The tests involved 34 participants divided into couples and randomly assigned to the platforms. The first setup involved a CAVE2 system and a *Playstation Move* controller mapped to a virtual wand. Users could see each other, use unrestricted full body and wands to interact, but only one of them was head-tracked (HT). This asymmetry resulted in the fact that HT users were more physically involved during the tasks. In the second setup users were sitting on a chair, tethered to a PC using *Oculus Rift* HMD and a *Leap Motion* sensor. They could not see each other, but their hands were showed in virtual environment. The tasks considered for the evaluation were basically two: finding shortest paths and counting the number of 3 vertex-cliques. Several extents were taken into account to quantify the impact on functionalities (e.g. accuracy, completion time), collaboration (e.g. strategies, shared focus, analysis of oral communication) and user experience, using a Likert scale based questionnaire.
It has been found that participants were highly accurate in both environments, but significantly faster with HMD. Display obstruction and the size of the room could have extended completion times for the CAVE setup. Even though HMD users were more independent they showed the same focus. No significant differences were also registered in terms of oral communication, perception of the partner and self-reported measures, suggesting that modern

HMDs, such as *Oculus Rift*, represent a comparable alternative to CAVE systems for collaborative data analysis and might even reduce the time required to accomplish the tasks.

2.1.8 Maps and globes in virtual reality

In 2018, Yang et al. [34] investigated the different ways to represent worldwide geographic maps in an immersive VR environment.
They compared, in particular, four proposed approaches: an *exocentric globe*, in which the viewer is external to a 3D sphere-shaped map, a *flat map*, where the map is a flat plane in VR, an *egocentric globe*, in which the viewpoint is internal to the globe, and a *curved map*, obtained by projecting the map onto the concave section of a sphere surrounding the viewer. In this scenario users could walk around the visualization using an HMD (e.g *HTC Vive*) and interact with the maps through a single shooting-beam controller.
Furthermore, authors provided an evaluation of these four visualizations in terms of user experience and efficiency, taking into account three task to be performed: distance and area comparison along with direction estimation. Several extents were considered such as response time, accuracy and the degree of interaction, recording the positions and rotations of the head for each user. 32 participants with mixed background on VR were involved.
The findings showed that the exocentric globe outperformed the other visualizations in all tasks. It did not cause any distortion and revealed to be the best for overall accuracy, leading to task execution times similar to maps and slower than them only for small area comparison. Flat map presented the whole world surface in the user's field of view, bringing good performances for comparison tasks, however the map distortion caused very poor results for estimation tasks. The egocentric globe represented the most immersive one, but perceptual distortion and the need for users to turn their head resulted in worst performances and highest perceived motion sickness. It was useful, though, to reveal small variations during comparison tasks. The novel curved map generally performed better than classical flat map, but lead to the second-worst motion sickness. It has been highlighted that, using exocentric globe, users preferred to interact using the controller rather than with head movements. This factor could have influenced performances.
Authors explained they focused on VR rather than AR due to the better field of view provided by current HMDs. They finally speculated on realizing, and eventually evaluating, hybrid transitional map-globe visualizations.

2.1.9 Origin-destination flow maps in immersive environments

In this paper from 2019, Yang et al. [35], extending the work made in [34], focused on spatial-embedded data representations in immersive VR environments, comparing several 2D and 3D visualizations for origin-destination (OD) flow maps.
Their research is built upon three user studies. As in their previous work, each one of them provides some findings based on the evaluation of user performances and self-reported experience.
In the first study, they compared two 2D (either using straight or curve lines) and three 3D (either using constant height for arcs, heights based on flow quantity and heights based on distance between locations) flows representation on a flat map. For every visualization user could both move in space using a VR HMD and pick the map using a tracked controller. No scale or filtering actions were provided. The required task was to find the greater flow among two pairs of locations. The main findings were that the 2D straight line flow map performed faster but was not so appreciated, while the participants tended to look on the side and to interact with the map more in 3D than 2D representations.
In the second study four visualization have been compared: 3D distance-based and 2D straight line (from previous study), a globe map and *MapsLink*, a novel approach using flow links between two flat maps. The interactions provided were the same of the first study, plus *geo-rotation*, allowing to change relative positions on the map. The task used for the evaluation was also the same. In addition to head movements, controller and map state, also the use of *geo-rotation* was recorded. 20 participants attended the tests. Findings mainly showed that *MapLinks* had the worst overall performances, *geo-rotation* improved results for the 2D straight line case, which also led to least head movements, and globe outperformed the others and had the most appreciated design.
The third study, finally, simply extended the second one to a significantly larger dataset, showing how the globe visualization appeared as the most accurate and fast one. Researchers were surprised by such performances, since participants could only see half of the globe at a time.
Final considerations involved the hypothesis to investigate collaboration in multi users environments and, again, the current superiority of VR HMDs to AR in terms of field of view.

2.1.10 FiberClay: sculpting three dimensional trajectories to reveal structural insights

In 2019, Hurter et al. [36] developed a tool, *FiberClay*, to display, interact and modify 3D trajectories, which are inherently spatial-embedded, in an immersive VR environment.
They focused on designing techniques to enable complex queries in this 3D space, which should allow the analysts to select and compare trajectories with specific properties. Through their analysis of requirements, researchers pointed out some general issues. Visual considerations involved the potential occlusion brought by the tangled nature of data, the need of scalability and the difficulty to perform accurate selections. VR considerations highlighted the controversial use of 2D GUI interfaces, the preference from users to remain seated while performing tasks and the need for multiple views.
This brought to design ad-hoc components. Controllers have been used as a bi-manual 3D brush. Users could draw lines to intersect segments of the trajectories, allowing to perform complex selections. To ease navigation, besides head movements, hand controllers could be also used to change the view, by translating, scaling or rotating it. Some specific attribute configurations were also mapped to a small grid placed on the floor. By moving on the grid using the joystick, these different configurations could be selected. The implementation has been built in C# using the *DirectX* API for a *Samsung Odissey* headset. Feedback from domain experts was also analyzed in four specific use cases: anomaly detection, traffic analytics, wind extraction and neuron fiber tracking. In every scenario, *FiberClay* has brought specialists who used it to gather interesting insights on data, proving to have increased the overall knowledge of the dataset. An important limitation in using the brushes has been observed when users tried to select far away objects, due to the amplification of hand movements. However, the grabbing interaction, used to bring objects closer, proved to be fast enough to mitigate that. It was also pointed out that the modern high-resolution VR devices alleviated the occlusion risk and the difficulty of making detailed selections. User reacted enthusiastically to the system, showing a high level of engagement. This could be due to either the novelty effect of such technologies or the actual effectiveness of immersion.
Future works on undo/redo techniques and multi-user collaboration have been suggested.

2.1.11 ImAxes: immersive axes as embodied affordances for interactive multivariate data visualisation

In 2017, Cordeil et al. [37] developed and presented an immersive tool to visualize and explore multidimensional data with fluid interactions: *ImAxes* (Immersive Axes).
The tool was built in Unity and has also been tested using an *HTC Vive* headset. The goal of researchers was to restrict the use of WIMP (Window, Icon, Menu, Pointer) interfaces, even though that usually constrain the flexibility of applications. Thus, every interaction has been built upon the direct manipulation of fundamental building blocks: axes, indeed. A formal grammar based on axis placement and concatenation has been developed to allow constructing traditional 2D visualizations (e.g. Histograms, Scatterplots, PCP) and their 3D variants. This has been done in order to have visualizations which look and behaves like physical objects. By combining simple axes, more complex and emerging visualizations, like overlaid plots and scatterplot matrices, could be obtained. Axes direct manipulation has been enabled by the means of 6DOF controllers and *grab and place* interactions, in which users press the trigger button while pointing at an object to move it and then release trigger to place it. Also gestures for *scaling*, *filtering* and *throwing away* were implemented. Also, moving plots one against each other as brushing wands, allowed to perform complex queries. It has been observed how users tended to place different visualizations around them in circular paths, within arms reach, and moved the attention by rotating in place, like in a workspace.
As mentioned above, also a test on a specific use case has been conducted. It involved a single data scientist who tried to gain additional insights over a dataset containing the features of thousands of wines. A first analysis focused on comparing white and red wines, while the second one was a quality inspection.
In both cases the user detected outliers and critical features, using *ImAxes* as an *embodied query* tool. However, it is said this simple scenario can not be seen as an effective evaluation. Other mentioned limitations are the number of displayable axes and data points (due to hardware restrictions), the lack of high interactivity in editing the plots and the need of more multivariate and multimedia data to enhance user perceptions.

2.1.12 3D visualization of astronomy data cubes using immersive displays

In 2016 a team of physicians and astronomers, in collaboration with computer scientists [38], built a multi-platform prototype tool to explore astronomic data, specifically spectral-line radio cubes from galaxies surveys, in a 3D environment.
Authors aimed to stimulate interest in their colleagues in embracing these technologies, especially new cheaper hardware like *Oculus Rift* and *Leap Motion*, to support their insights. The prototype was developed with Unity game engine for CAVE, *zSpace* tabletop, desktop and HMD. They first converted astronomic data from a FITS format to binary, using Python, and then fed them to Unity, which read them as a single 3D texture. The texture was then applied to an actual cube placed on the scene. It has been pointed out how volume rendering is demanding for GPUs. Just few basic interactions with the cube were possible: translate, rotate, scale and slice.
They did not provide any proof of concept nor any evaluation, but foresaw many perspectives for future improvements. More natural interactions with cubes, such as showing details on hovering, were suggested, along with the possibility to overlay multiple data, like 2D floating panels. It has been noted however that a useful tool for science should allow to *explore, explain, extract.*

2.1.13 A study of layout, rendering, and interaction methods for immersive graph visualization

In 2016, Kwon et al. [39] made a study on representing abstract data, particularly graphs, in an immersive environment, driven by the limitations of 2D techniques and the growing popularity of HMDs.
Their basic idea was to place traditional 3D graph layouts on the surface of a sphere, placing then the user at the center of the sphere, in order to reduce occlusion and allowing an overview of the nodes. Authors also focused on domain-specific topics like depth routing techniques, which bring more intricate bundles closer to user's viewpoint, and optimized graph rendering but also interaction in such an environment has been discussed.
Despite the growing popularity of novel interaction techniques in VR (e.g. *Leap Motion*), researchers decided to adopt a traditional cursor approach. After having discussed, in a preliminary study, the different setups obtained combining user's view with mouse, they decided to follow a classical mouse-only

cursor paradigm, providing also a key shortcut to reset the cursor's position at the center of user's view. Only three basic interactions were designed: pointing, highlighting (left-click) and selecting (right-click). The nodes highlighted through the cursor were brought closer to the user. Also the nodes adjacent to the selected area were brought closer, but not more than the selection itself. This resulted in highly distinguished nodes and the possibility to easy follow connections to their ends.
Authors also conducted an user study with an HMD to prove the effectiveness of spherical layout and depth routing for graph visualization. They provided an evaluation based on 4 tasks performed on 3 different graphs in 3 display conditions. Specifically, 2D layout, spherical layout without depth routing and spherical layout with depth routing were compared. Tasks were about finding paths and interesting nodes. Extents like completion time, correctness rate and number of interactions have been taken into account. 21 participants attended the tests and their feedback was also reported through questionnaires. Findings showed that users preferred and performed better with the spherical layouts and that depth routing significantly helped in accomplishing tasks. 2D layouts performance were strongly influenced by field of view issues, which led to occlusion and clutter. It was finally noted how HMDs limit the use of traditional input devices, like mouse, and that 6DOF controllers will be preferable in the future, since they are becoming ever more precise and affordable.

2.1.14 VRMiner: a tool for multimedia database mining with virtual reality

In 2006, Azzag et al. [40] investigated a new 3D interactive method for visualizing multimedia data with virtual reality and built a related tool, named *VRMiner*.
Considering domain-specific databases, researchers thought to represent data as a cloud of objects in an immersive environment, to let the domain experts explore them, stimulating the finding of several correlations. Hence, they encoded the data numerical and symbolic features as graphical attributes like object's position, color and shape. Furthermore, the support to more complex multimedia attributes such as images, videos, files, sounds and 3D models, has been considered, in order to enrich perception from users.
To achieve these purposes, the system has been designed to be used along with a pair of LCD glasses (due to the high cost and low performances of

stereoscopic HMDs in that period), a data glove to interact with the objects (by the means of few simple gestures), a tracked ring sensor to navigate (by zooming and moving the camera view) and a second computer, on which related images, videos and web-pages could be showed in detail (following the "details-on-demand" paradigm). Objects could be pointed and selected either using the glove or the mouse. Authors kept the tool as general as possible to make it suitable for different database domains. It has also been provided the option to record the user's navigation as a video. In order to avoid perspective distortion and to give informative value to the different size of the objects, it is suggested to introduce an invariant property for each one of them (e.g. using pyramid shapes with the same basis but different heights). Moreover, it is highlighted how the detailed information on the second computer made the tool a sort of advanced browser.
Authors also realized an user study, testing the tool with 15 real databases regarding skin analysis. The system was revealed to be effective in detecting correlations and outliers, checking the data distribution and presenting details to a team of experts. Users found 3D displays easier and more efficient than other standard 2D softwares.
Important drawbacks have been seen in the initial difficulty experienced from users in understanding how data features were mapped to graphical attributes and in the lack of an interface to modify such mapping (except the space coordinates). Interesting perspectives were also seen in using more advanced controllers (e.g. wands) and in comparing the tool with other existing ones.

2.1.15 The hologram in my hand: how effective is interactive exploration of 3D visualizations in immersive tangible Augmented Reality?

In this article from 2018, Bach et al. [41] realized a comparative user study on three visualization technologies for 3D environments: an augmented reality HMD, i.e. *Microsoft HoloLens* (*ImmersiveAR*), an handheld tablet, i.e. *Microsoft Surface* (*TabletVR*), both along with a tangible paper marker used as interaction tool, and a desktop (*Desktop*) setup.
For this purpose, three aspects were studied: stereoscopic perception, degrees of freedom in interaction and the role of physical proximity in perception and interaction. Authors analyzed four tasks performed by 15 participants on a 3D

cloud of points placed inside a cube: distance estimation, cluster detection, selection of an element and orientation of a cutting plane.
The desktop setup consisted of a standard monitor and a mouse, which could be used to rotate the view. Only left-click was required for each interaction.
In the tablet setup users were free to stand or sit, with both hand and visualization appearing behind the screen. Together with the paper marker, a cardboard with glued markers was used to render the visualization.
The same markers were used also in the HMD setup. Users could stay either standing or seated and use the *HoloLens* clicker along with the interaction tool.
The only provided interactions were rotating and selecting. Users could rotate the visualization (by mouse drags or marker rotation) or walk around it (*TabletAR* and *ImmersiveAR*). Selection could be made through clicking (*Desktop*), touching the screen(*TabletAR*) or moving the marker to place a 3D pointer (*ImmersiveAR*).
For each task completion time and error rate were considered as extents. A questionnaire for participants was also used in order to rate their experience. In addition, 6 random individuals from the sample were trained for five consecutive days, representing the so called *long-term training group*.
The study findings showed that *ImmersiveAR* outperformed other visualizations in selection and cutting plane tasks, requiring direct manipulation, and performed at least as well as *Desktop* in terms of accuracy in any other task. *TabletVR* had the worst overall performances. *ImmersiveVR*, though, led to the slowest performances because participants took extra time to verify their answers, to explore the holograms and due to the novelty of technology many of them were inexperienced. Training was pointed out to alleviate these drawbacks. In general, *ImmersiveVR* was perceived in a very different way from users, proving that this technology is tightly related to individual subjectivity. *Desktop* showed overall good performances, maybe due to the minimal effort required to interact with it and to the fact that all users were already familiar with it. It has been pointed out that users complained about the restricted field of view in *ImmersiveAR* and that the requirement of staying seated could have led to faster performances.
Several limitations were tightly related to the AR technology, such as slightly lagging markers, the difficulty of accurately orientating them and the reduced view quality of holograms when moving in close proximity.

2.1.16 Multilingual semantic cyberspace of scientific papers based on WebVR technology

In 2018, Charnine et al. [42] realized an immersive visualization of scientific papers in a VR environment using a WebVR approach, in order to foster dynamic learning and the discovering of new articles, ideas and trends.
Papers have been represented as spheres and the more significant and cited they were, the larger these spheres will be. Starting from a corpus of documents, a matrix of mutual similarities among each paper has been computed. Then, a dimensionality reduction method called *t-Distributed Stochastic Neighbor Embedding* (t-SNE) has been used to obtain 3D coordinates of spheres from the matrix. As mentioned, the size of each sphere was tightly related to its significance, so a semantic similarity measure (called *Science Contextual Citation Index*, SCCI) has been proposed as well, to provide an index of how much an article is scientifically significant and related to others. The more two articles were correlated, more similar will be the color assigned to the respective spheres. Researchers computed this similarity considering not only the number of bibliographical references among two articles (as done by the usual *Science Citation Index*, SCI), but also implicit links, i.e. the common text fragments, in order to take into account also texts with no formal references. Differences among multilingual articles, e.g. Russian and English, have been taken into account as well.
A-Frame has proven to be a powerful WebVR framework, translating these data into a 3D scene with few HTML statements. Nevertheless, no interactions nor filtering actions were provided and the system was not formally evaluated.

2.1.17 Development of emergency drills system for petrochemical plants based on WebVR

In this article from 2011, Chun et al. [43], combining together Web and VR, realized an immersive training system for firefighters to handle petrochemical plants emergencies. Firefighters could easily access to a virtual simulation of the plant and move freely, test the equipment, conduct drills or watch some instructive video clips.
It has been pointed how WebWR has increasingly overcome its previous limitations, becoming more and more vivid and performant. The system has been realized in Visual C++, using also a fire dynamics simulator (FDS) and 3D

models of the plant in 3DS format. The architecture was based on the *accident simulation module*, to simulate plants and accidents, the *emergency response module*, to perform drilling and training, and a 3D web virtual reality engine, to perform real-time interactive tasks (e.g. using a foam extinguisher). The functionalities provided included the possibility to explore the virtual plant using keyboard and control stick, fire simulations in different weather conditions integrated with multimedia data like images and texts, possible adjustements on fire equipment and a cooperative mode in which users can collaborate in different roles.
However, no evaluation of the user experience has been provided. Even though if this work is not strictly related to the Immersive Analytics area, it is worth to highlight how the WebVR approach has also enabled collaboration and an easy propagation of eventual plants modifications to each user, since the system is based on a simple client-server architecture.

2.1.18 Exploring data in virtual reality: comparisons with 2D data visualizations

In 2018, Millais et al. [44] compared VR and 2D visualizations in representing multi-dimensional datasets, taking into account the workload and the derived insights. In particular, they realized two VR visualizations and their equivalent versions in 2D.
In the first setup, called "*Be The Data*" users were immersed in a three dimensional scatterplot, in order to examine data from an internal perspective. Environment could be freely navigated by the means of controller, points could be hovered to show their details and, as the most interesting feature, by clicking on them users could "become the point", viewing the dataset from the point's perspective.
The second setup, *Parallel Plans*, represented different dimensions as evenly spaced plans, linked by spline lines in correspondence of each point. In this environment, users could move freely around and hover on specific points to retrieve details, as well. Both the VR visualizations have been built using Unity and tested by the means of *Google Daydream* HMD and controllers. On the other hand, their 2D counterparts have been realized using HTML, CSS and D3.js and provided same interactions through simple click actions.
Researchers further conducted an user study to compare the four setups, aiming to quantify workload, precision and satisfaction. 16 participants attended

the tests and they were required to communicate their insights as they were thinking aloud.
Even though the workload has been found to be overall the same, users rated their performances in VR as less fatiguing. This can get even more interesting if we consider that some of them used the VR tools while standing, whilst they remained seated in the 2D setups. Moreover, participants reported fewer errors in VR rather than in 2D. Authors observed that many users did not even use the hover functionality to provide details, as they deduced points' coordinates by simply observing them. This approach led them not to find deep insights. A curious finding is that most of the users ignored the "becoming the point" functionality, mainly inspecting the datasets from a frontal view, thus neglecting the third-dimension benefits.
Perspectives included implementing collaborative features for the tools.

2.1.19 The data in your hands: exploring novel interaction techniques and data visualization approaches for immersive data analytics

In 2018, Hube and Müller [45] performed an exploratory study on both novel interaction and visualization techniques in immersive VR environments. Specifically, they focused on *Parallel Sets* representations, suited for multi-dimensional data, in which objects are plotted as polylines connecting parallel plans, each one representing a dimension. An early designed traditional 2D approach was then enriched with 3D spatial features, allowing users to switch among these two visualizations.
Authors studied a specific use case: analyzing sales and revenues. Even if 3D has been said to lead to several drawbacks as occlusion, inaccuracies derived from perspective and overlapping paths, it could bring some benefits in revealing complex structures related to interesting insights.
Several features have been provided inside the application: besides switching the visualizations, users could walk inside it; in order to visualize different versions of the data, multiple views could be overlaid; users could also move freely inside the visualization, interact with points and paths for filtering purposes using controllers and zoom/rotate the view with simple gestures. Furthermore, a novel hand-attached menu, which users can interact with using lasers cast from controllers, has been designed to achieve more complex tasks. It could provide a complementary view of the dataset as a 3D scatterplot and

summon previous axes configurations as thumbnails for comparing purposes. The prototype has been realized for *Oculus Rift* and *Oculus Touch* controllers, using the *Blueprint* system in Unreal Engine 4, which supports C++.
It has been pointed out that exploring from a close perspective might lead users to lose the overview on the visualization, consequently supporting tools (e.g. a map) should be provided. Also, additional navigation modes have been suggested to easily surf the data. It has been noted how transitions between 2D and 3D visualizations could disorientate users. Finally, authors speculated on providing support to multi-user collaboration, allowing to share different views of the dataset.
No formal evaluation for the tool has been provided.

2.2 Related Works

Here in the following are discussed the previous studies carried out by the DEI research group that established a basis for the present investigation.

2.2.1 A taxonomy generation tool for semantic visual analysis of large corpus of documents

In this article, Onorati et al. [46] described a tool for both the generation and visualization of taxonomies, in order to categorize unstructured datasets and improve knowledge over them.
Taxonomies, though, are domain-specific and require experts to be validated. Such tool was meant to support domain experts, hence a semi-automatic approach has been used to *gather* a corpus of documents, *extract* the relevant terms and *categorize* them into semantic groups. Indeed, experts might still contribute in each step, by modifying the uploaded documents, the extracted terms (e.g. setting a different frequency threshold) or the created categories (e.g adding or removing terms), bringing more value and meaning to the taxonomies.
A NER (Named Entity Recognition) tagger was firstly used to identify terms with special meaning (e.g. people, institutions, events) and their categories. Then, terms were also grouped together if they were semantically related. Four semantic relations have been considered: synonyms, antonyms, co-occurrences (i.e. terms appear one in the dictionary definition of the other), and multi-word

terms. The taxonomy could also be saved as a hierarchical JSON file. Furthermore, at the end of the process a visualization of the taxonomy was automatically generated. Categories were represented as colored circles containing other circles, representing terms. Zooming was provided and by clicking on terms, a list of the documents in which they appeared was shown. It has been pointed out how such a taxonomy can be useful in three scenarios: to support visual analytics, to serve as an interactive dictionary and to perform semantic searches. To guarantee portability among different devices and operating systems, the tool has been implemented as a web application.
Moreover, an usability evaluation on two use cases, politics and biotechnology, has been carried out involving 16 participants with no previous modeling knowledge. During the evaluation they were asked to fill a pre-test questionnaire, then to perform some tasks with the tool (e.g. upload a document, sort terms, edit categories) and to fill a second questionnaire after the test.
Findings showed that the tool was easy to use even for not experienced users, who appreciated the possibility to interact both with buttons and direct manipulations (e.g. drag and drop). It has been mentioned how an immersive visualization of generated taxonomies, isolating users from reality, could stimulate the exploration of the dataset, fostering learning of advanced topics even for non-expert users.

2.2.2 Designing an immersive visualization for semantic datasets

Onorati, Diaz et al. [47] kept investigating on taxonomies and semantic analysis, developing an immersive visualization in VR surrounding users with terms and categories obtained with the approach mentioned above. Besides the Immersive Analytics main opportunities, their aim was to exploit the novel and cheaper VR technologies to gain knowledge over large datasets, in order to overcome the spatial limits of a 2D visualization.
The data were modeled as bubbles (i.e. spheres) whose size was related to the weight inside the corpus (for terms) and the number of contained elements (for categories), similarly to a traditional 2D bubble chart. The prototype has been developed using Unity for *Oculus Rift* and *Oculus Touch*. Users could navigate the environment and interact with the bubbles through the ray-caster controllers. Bubbles could be naturally grabbed and moved around while users

were allowed to perform exploration tasks like entering categories and add/remove terms from them using natural gestures. Since the taxonomy datasets are inherently hierarchical, it has been also provided a *breadcrumb* on the wrist to keep track of the currently visited views.
The tool was tested on a specific use case: tweets from 2017 Catalan crisis. To maintain a journalistic approach, five more categories were added to the semantic analysis and used as initial bubbles: *who, where, how, what and when.*
An exploratory study on the tool has been carried out at ACM UIST 2018, involving 11 researchers from virtual and mixed reality. They were asked to try and explore the tool and then filled out a questionnaire based on a Likert scale.
Participants reported to feel absorbed from the experience and to have gained knowledge over the data. Furthermore, significant improvements were recommended, like the possibility to see previous scenes in transparency through the bubbles and to guide users with visual suggestions. The study showed that this kind of application created an interest in users, fostering researches on the benefits of immersion for semantic analysis.
Future perspectives have been seen in modifying interactions and introducing some kind of visual mechanism to filter out elements.

2.3 Considerations

This thesis aims to deepen the topics discussed in [47], realizing a similar tool that overcomes previous limitations and offer novel features, taking into account the results of the exploratory study, the current IA agenda and the different solutions previously examined. Furthermore, following the idea presented in [46], this new application should also enable domain experts to actively correct the result of the preliminary taxonomy.
In Table 2.1 are summarized the features used to categorize researches in IA, the works in which those features have been provided, the approach used in [47], and, lastly, the choices taken for this thesis work.

Table 2.1: Summary of the different features encountered in IA solutions and comparison with previous work. Novel contributions are marked with (*).

<table>
<tr><th>Feature</th><th>Different options</th><th>Works that use this approach</th><th>Approach used in previous work</th><th>Approach used in this work</th></tr>
<tr><td rowspan="2">Technology</td><td>VR</td><td>All works but [41]</td><td colspan="2">✓</td></tr>
<tr><td>AR</td><td>[41]</td><td colspan="2"></td></tr>
<tr><td rowspan="2">Data Type</td><td>Spatially-embedded</td><td>All other works in literature</td><td colspan="2"></td></tr>
<tr><td>Abstract</td><td>[30], [32], [37], [42], [44], [45]</td><td colspan="2">Semantic hierarchical data</td></tr>
<tr><td rowspan="5">Input type</td><td>Mouse and keyboard</td><td>[28], [29], [30], [32], [39], [40], [41], [42], [43], [44]</td><td></td><td>✓</td></tr>
<tr><td>Controller, wand</td><td>[26], [27], [28], [29], [30], [33], [34], [35], [36], [37], [43], [44], [45]</td><td>✓</td><td>✓</td></tr>
<tr><td>Touch, hands</td><td>[29], [33], [38], [40], [41]</td><td></td><td></td></tr>
<tr><td>Gaze</td><td>[32]</td><td></td><td>✓</td></tr>
<tr><td>Voice</td><td>Not mentioned</td><td></td><td>✓(*)</td></tr>
<tr><td>Perspective</td><td>Data in hand</td><td>[26], [28], [32], [33], [34], [35], [36], [37], [38], [41], [45]</td><td colspan="2"></td></tr>
</table>

Continued on next page

Table 2.1 – continued from previous page

Feature	**Different options**	**Works that use this approach**	**Approach used in previous work**	**Approach used in this work**
	Inside the data	[27], [29], [30], [39], [40], [42], [43], [44]	✓	
Navigation	Walk around	[26], [27], [29], [32], [33], [34], [37], [38], [39], [41], [42]		
	Scene manipulation	[35], [36], [40], [41], [45]		
	Continuous navigation	[28], [30], [43]	✓	
	Fast navigation	[44]		✓(*)
Interaction type	Manipulation	All works but [27]	✓	
	WIMP	[27]		
	Both	[43], [45]		✓
High-level tasks	Visualizing	All works	✓	All of them
	Filtering	[28], [36], [37], [39], [41], [45]	✓	
	Correcting	[26], [43]		
Collaboration	Synchronous	[29], [33], [43]		
	Asynchronous	Suggested in [26]		✓(*)
Supported devices	Desktop	[28], [29], [30], [32], [38], [40], [41], [42], [43]		✓
	CAVE-like	[27], [33], [38]		
	HMD	All works but [27]	✓	✓

Continued on next page

Table 2.1 – continued from previous page

Feature	**Different options**	**Works that use this approach**	**Approach used in previous work**	**Approach used in this work**
Additional data	Text	[29], [43]	✓	Text and images
	Multimedia	[40], [43]		
Type of application	Stand-alone executable	All other works in literature	✓	
	Web distributed	[29], [32], [42], [43]		✓
Evaluation	Usability	[36], [37], [40]	✓	
	Comparative	[26], [28], [29], [30], [33], [34], [35], [39], [41], [44]		✓

In the first place, it is interesting to note that most of the researches focused on VR rather than AR. The former technology has proven to deliver a more immersive experience, leading to fewer distractions [35] and to a wider field of view [34] than the latter. Plus, VR technologies are improving their performances at a faster pace, while AR marker tracking may still be enhanced [41]. Such differences are also reflected in the cost, which makes VR HMDs significantly more affordable. For these reasons, also in the current project, VR has been chosen over AR.
Basically, all the solutions can be grouped into two categories according to the type of data on which the proposed visualization is based. The first category involves physical [27, 43], astronomical [38], topological [33, 39], biological [26], chemical [28], geographical [34, 35, 36] data, i.e. all those cases in which information has an inherent spatial component. On the contrary, the second category includes purely abstract data, like multi-dimensional and, indeed, semantic data. Such information, in fact, has no trivial concrete representation and clever solutions (e.g. PCA [30], parallel plans [45]) need to be found for each specific situation. Anyway, research has kind of neglected this last domain in favor of the first one.

Another distinguishing factor for IA solutions is input. Along with the traditional mouse and keyboard setup, which is still provided in the majority of works and has been explicitly preferred in [39], several tools supported tracked 6DOF controllers or wands. These can be used essentially as raycasters or gesture-driven manipulating entities (e.g. abstracting virtual hands), enabling intuitive interactions. Furthermore, by the means of sophisticated sensors, like *Leap Motion* [29], data gloves and tracked rings [40], or through the use of touch interfaces [41], the user's physical hands can be rendered within the virtual environment. Also the use of a gaze-based cursor has been experimented [32], proving to be useful for showing details while hovering. It is important to highlight that, on the other hand, no examined solution provided support to a vocal input, even though it is mentioned as one of the possible ways to interact in [22]. Speech, therefore, can represent an alternative method to deliver textual input in a virtual environment. It can also be exploited to query data more quickly and naturally. While the previous work supported only tracked controllers, this thesis intended to explore all these possible approaches. Unfortunately, hand-tracking and touch input require particular hardware (e.g. touch screens, motion sensors) and related software interfaces that were not available for this research. By the way, modern tracked controllers (e.g. *Oculus Touch*) can accurately render user's hands in VR, offering a powerful and affordable alternative mean for interacting through natural movements.

It has been found that all solutions essentially provided two perspectives on data. The first, and more common, perspective is the one in which the whole information is reported as a manipulable object (e.g. a neuronal trace [26], a scatterplot [30], a world map [35]) at the user's hand. This approach is particularly useful to have a general overview of data but can become cumbersome when it comes to analyzing details. A second perspective is the one in which users are placed directly inside their data, surrounded by entities (e.g. in a multi-dimensional cyberspace [29, 42], within a scatterplot [44]). This choice has proven to be helpful to gain insights [39] and find outliers [44], but it is more likely for users to get lost and disoriented. Thus, this choice must be associated with components that support navigation (e.g. a mini-map [26]). For this purpose, the previous application provided a *breadcrumb* menu on the wrist to keep track of the visited views[47]. Such a menu has been appreciated by users and was included in the current version too. It has been complemented, though, with several text labels placed right in front of the virtual camera. These labels are located at the margins of the visual range, in order not to block the view, but rather supply the same information on the previous inspected levels. Additionally, they show the name of the taxonomy file used to generate the model, report the recognized speech, and give useful instructions to guide users through tasks. In this way, such details are always available at sight, so that there is no need to look down on the wrist.

Several ways to navigate the virtual environment have been described, but they can be grouped mainly into four categories. Most tools did not provide an actual navigation system, allowing users to simply walk around the scene. Nevertheless, this choice is effective only in room-scale systems [27] or using wireless devices [41]. When provided, this feature has not been so used, while users chose to stay seated and to bring objects closer to them [26, 34]. A second way to navigate was provided in some works by allowing the users, rather than moving themselves, to scale, zoom, and rotate the view of the scene. Such actions have been also implemented through hand gestures [45]. This approach is better suited when data are represented at hand but can be neglected when the user is surrounded by them [40]. A third possibility is to provide continuous navigation mechanisms, i.e. simulating the movement of users in the environment through keyboard [30] or controller thumbstick [28, 43]. The previous version of the application belongs to this latter category, as well. This technique seems more natural and is also often used in many games, but it is more prone to cause motion sickness [30]. When performing such navigation, users reported discomfort due to the high velocity of the simulated movement and a low, not adequate frame rate. Hence, considering the user predilection for sitting, without forgetting the necessity to explore from different angles and the need for fast movements, this thesis will use a fourth approach. Inspired by the *"become the point"* interaction described in [44], it has been implemented a rapid way to move around, called *teleportation*, by pointing and clicking with the laser ray on the desired location on the ground. Even if this feature is provided in several virtual game environments, it has not been encountered in any of the examined works. It should allow reaching further destinations in an instant, overcoming the limitations of a continuous transition. Besides, the above-mentioned *breadcrumb* menu has been further enhanced, making it interactive, so that it can also be used to fast travel among different levels of the semantic model.

In the large majority of works has been pointed out the importance of natural interactions with virtual elements, as well as it has been avoided the use of traditional menu interfaces (WIMP) [37]. Just the early work described in [27] presented them as the only way to give commands. Some solutions also mapped a set of gestures to complex contextual actions [40, 36]. This choice is consistent with the concept of *embodiment* [22], but it has been observed how it is restricting in terms of flexibility of the application [26, 40]. In some cases, including the previously realized tool, users struggled to remember the various combinations of buttons and gestures [36, 47]. Plus, a graphical menu becomes necessary when it comes to navigating the functionalities of a system [43], to provide additional views or to iterate through previous configurations [45]. In the same way, the present work will try to minimize the number of graphical menus in favor of direct manipulation, but it will provide them to perform complex actions on bubbles and to act on the whole scene. In particular, starting from the idea presented in [45], hand-attached menus have been adopted. In addition, each bubble has its own contextual menu. All menus will be shown/hidden on-demand to reduce users' cognitive load [26].

Concerning high-level tasks, although every work provided visual support to the analysis, only a few of them allowed to interact with the environment to filter elements, e.g. hiding some of them [28], highlighting some others [39, 36] or acting upon attributes of interests and thresholds [37, 45]. Similarly, the previous work allowed to relocate bubbles inside categories, and to move them around, as a way of visually filtering them [47]. Nevertheless, just two researches considered the option to apply corrections on the underlying data[26, 43]. In this thesis all these three features will be provided by building the virtual model, filtering elements through interactions (e.g. hiding, moving) or vision (e.g. colored indicators), and editing them (e.g. adding, deleting, renaming entities).

Even if it is mentioned in many researches as a future perspective [34, 36, 45], only a little has been done to bring support to collaboration. The previous version of the bubble chart did not consider this aspect either. The most interesting proposals in this regard were all synchronous, like a live chat [29], a shared view [33] and the possibility to join the visualization playing different roles [43]. However, there is no solution provided for asynchronous collaboration. Only in [26], authors have speculated about the opportunity of editing previous sessions. Starting from this idea, it was decided to let the users store and share the taxonomy files associated with their exploration. It has been implemented through one of the above-mentioned hand-attached menus, using an intuitive save-and-load interface. This represents, in fact, a mechanism to

support asynchronous collaboration which has not been encountered in the examined literature.
The entire field of IA has gained raising interest just due to the evolution and diffusion of modern HMDs, so it is not surprising that almost the totality of solutions make use of them. This work, just like the previous one, will likewise focus on them as primary devices. In addition, since the traditional desktop setup is the most widespread and familiar one, though, a version for it with reduced functionality has been realized too. CAVE systems, on the other hand, remain extremely expensive and a privilege of few labs. Besides, several works like [33] have proven that probably their cost is not even justified anymore. Due to these reasons, such complex systems have not been taken into account.
Interestingly, few works supplemented and embellished their visualizations with additional context information employing articles and multimedia. Beyond [43] (not exactly an IA tool) that uses papers, images, and videos to add value to the simulation, only [40] offers an interesting contribution, exploiting images, videos, 3D models, and sounds to code several dimensions of multivariate data. The platform built in [29], instead, provided for all data objects external textual descriptions accessible through clickable hyperlinks. The previous version of the tool showed, for each word, a list of tweets related to it [47]. Similarly, this thesis aims to exploit additional objects to enrich the semantic model representation. For each term in the corpus, indeed, related texts and images will be extracted from a catalog file and represented inside the corresponding bubble. These details should immediately recall to users the meaning, or the semantic field, of a word (e.g. pictures of plants and farmers will be shown inside the term *agriculture*, DNA spirals for *gene*, clocks and calendars for *time*, etc.), and show some significant press articles on the topic. Such extra information might increase engagement and speed up the learning process over the dataset.
As one can easily see, concerning the building of the discussed solutions, the actual trend is to use a game engine (mainly *Unity*) and to deploy the system as a stand-alone executable. The earlier tool was developed following this approach too. An alternative methodology consists in distributing the application across the web. This second approach was chosen to foster accessibility, portability, and collaboration.
Also Unity can be exported to the web, as has been done in [29], which also highlighted how running in a web browser could be appreciated by those who

are skeptic about game-like environments. Another option is using *WebVR* libraries and frameworks, and realizing the application through web pages, with no build step. So a decision had to be made among these two technologies. Basically, both alternatives are valid and it is mainly a trade-off between stability (Unity) and speed in development (WebVR). In the end, it was decided to use WebVR and, specifically, the A-Frame framework. The comparison and the reasons for the choice are discussed in detail in Section 4.1.2.
Either way, the web approach constitutes a significant enhancement from user's perspective, since, as it has been pointed out, requires no extra software installation, still it can provide a vivid and immersive visualization with a good frame rate, the application can be reached just by using an URL and it can be tested even on a desktop setup [32, 42, 43].
Some of the examined researches also provided an evaluation to prove the actual effectiveness of the related work. Two kinds of evaluation can be identified. The first is an usability test, in which the application alone is assessed. The previous work provided such an evaluation. The second kind is a comparative test, involving the discussed tool along with similar visualizations. This second type of evaluation has mainly been used to quantify the benefits of immersive visualizations over traditional 2D applications [26, 29], other 3D visualizations [34, 35, 28] or other immersive environments [33]. In both kinds of evaluations are estimated the user performances, through a set of tasks, and user experience, using a Likert scale questionnaire. Several extents are usually considered like completion time, error rate and the number of interactions. Similarly, for this project, a comparative evaluation of the user performance and experience using the application in two different conditions has been carried out after the development phase. An immersive setup and a non-immersive one were assessed. As mentioned, this will be further discussed in Chapter 6.

Chapter 3

Requirements

The basic idea for this thesis was to realize an application which, reading the aggregated semantic data in a specific format (i.e. JSON), populates a virtual environment with entities tied to those data.
As mentioned in Section 1.3, this is part of broader research led by the DEI team at University *Carlos III* of Madrid and it has been conducted in collaboration with them. The members of the team, thanks to their previous experience in the fields of Human-Computer Interaction and Information Visualization, taking also advantage of the evaluation described in [47], delivered the requirements for the present work. Such requirements, which guided the subsequent technology adoptions and development process, will be discussed in this chapter here below.

1. First of all, it has been emphasized the need for *portability* over different devices, allowing several benefits both for the end-users to access it and for the researchers to test it during the development. Indeed, the heterogeneity and still limited availability of modern HMDs had to be taken into account and should not be an obstacle for the fruition.
   Thus, a web-distributed cross-platform version of the application has been realized. It is easily accessible via URL using both a stereoscopic headset and a traditional desktop setup, eventually with reduced functionalities. The majority of modern HMDs are supported.

2. Plus, the domain-specific essence of taxonomies had to be considered, keeping the tool *general*, applicable to any semantic dataset. This implied that the virtual environment could not be designed for a specific use case

(e.g. politics, healthcare, biology, etc.) using specific 3D models, backgrounds or sounds, but had to abstract the low-level details. Moreover, no assumptions could be made on the maximum depth of such hierarchical clusters, allowing users to move back and forth through an indefinite number of layers.
Hence, the idea of using self-containing bubbles, although simple, confirmed itself to be still very expressive for this purpose.
Further, the application code has been developed to generate each level of the visualization with the same generic algorithm. The only differentiation is about stepping into a category or a term. In the first case, other bubbles will be recursively created, while in the latter term-specific details will be shown. In this manner, the application will likewise behave independently from the domain and hierarchical structure of the input taxonomy.

3. According to the principles of Visual Analytics, it has been recommended not to overload users with too much information during the exploration and to show *details on-demand*.
For this reason, it has been decided to supply them with a progressive retail level only if they show interest for a specific term or category (e.g. looking at or bringing it closer), reducing to a minimum the direct display of digits and numerical data. It was done to foster a preliminary overview of features and relationships.
The previous tool showed a list of tweets related to terms for all of them at the same time. Unlike it, here detailed information concerning a term (e.g. pictures, descriptions, articles) is shown only when the corresponding bubble is entered.

4. As discussed, an immersive analysis tool relies heavily on *intuitive interaction* techniques to make the experience engaging.
This requirement led to the choice of using direct manipulation of data (e.g. moving entities around, grabbing, pulling them apart) to enable both filtering and navigation actions. Besides, using the previous tool, users showed some difficulties in remembering tricky combinations of buttons, so it has been suggested to include more traditional, but still easy to use, interfaces to accomplish delicate tasks like deleting and renaming entities.

5. Further, it has been pointed out the need for an *effective navigation system*, which allows users to easily and rapidly explore the data. Since there is no limit to the number of nested levels, there had to be a functionality to move fast from a view to another, eventually skipping the intermediate ones. Similarly, so as they do not get lost in the tangle of terms and categories, a mechanism to constantly remind them their current path had to be implemented.
   In the previous tool, navigation was based on controller thumbsticks, not allowing skipping, while a virtual display (*breadcrumb*) on the right wrist showed an overview of the user path.
   In the present work, instead, these issues have been addressed by implementing a locomotion system based on *teleportation*, an interactive *breadcrumb* menu that enables quick traveling back to visited levels, and text boxes attached to camera providing summary information.

6. It has been required that the *visual* design of each scene must stimulate *insights* at a first glance. Just by looking around, users should be able to detect the most relevant entities, in the same way as they should detect outliers.
   This immediately brings to mind to use the graphical features of the objects to model their numerical attributes. Previous researches like [40], though, have shown that, when elements are spread throughout a 3D environment, perspective leads to a distorted perception of relative sizes and distances. Therefore, it has been also recommended to find some visualization solutions to address this issue.
   Like in the other tool, the radius of categories has been modeled according to their importance inside the taxonomy, while here terms have all the same volume. As a novelty, level bars were integrated into the latters as discriminant components. Bubble labels, besides, are all of the same size, alleviating the perspective issue.
   The exploratory study on the early tool has also brought out the need from users, when inside a certain bubble, to be able to see what was on the previous level, suggesting to use a transparent background.
   This advice has been precisely followed, surrounding users with a large see-through semi-sphere when in an inner level, placing previous level bubbles behind it.

7. It has been requested to allow users to actively *correct* the semantic model

used as input for the visualization. This allows not only to manually intervene on the output of the automatic categorization process but also to group new and pre-existing entities in a custom way that facilitates future revisions. Specifically, as it was in the 2D tool described in [46], they should be able to add, modify, delete or relocate terms and categories.
Only the latter feature was available in the early work, while all the others have been implemented by this thesis project. Contextual menus attached to each bubble have been designed to accomplish these tasks, eventually requiring also natural gestures (e.g. point-and-shoot) to make them less cumbersome. As mentioned above, relying only on gestures could result in discomfort when it comes to perform sensitive actions.

8. The need for experimenting with new interaction techniques, easing the navigation through the environment, and making the exploration more absorbing, resulted in the necessity of designing a way to implement *vocal inputs*.
   Besides the fact that previous works lacked this novel feature, it constitutes a powerful tool for inserting strings in the application when a keyboard is not available. Plus, it can be used to quickly give navigation instructions. Extending this concept, using voice might also be the fastest method to find an entity during exploration. Thus, speech recognition has been included to support all these mentioned functionalities.

9. A support to *related data objects* has been required to enrich the environment with accessory representations of a term.
   As said, about this aspect, the previous solution relied on collections of tweets. For this visualization, rather, it has been recommended that relevant articles, images and eventually videos can be helpful for individuals to picture the concepts in their minds, to give them a clear overview of that specific topic, and to learn new facts concerning it.
   Since the virtual environment can abstract a gallery, such 2D data were easily integrated into a term bubble using standard solutions like floating panels. However, some visualization guidelines were taken into account to minimize occlusion and distortion, e.g. using high-contrast colors and inclining content toward the camera.

10. Furthermore, a way for users to *collaborate* had to be offered.
    Since a single session with the application could be not enough to carry

out the desired corrections, it has been requested to implement a functionality that enables users to save and share the changes they applied to the model. This should allow, at the same time, to inspect and modify someone else's view on data and to resume an already commenced exploration. Moreover, such function may reveal to be useful also to evaluate the designed prototype, since the saved sessions constitute permanent records of the users' activity.
This facet was not even considered for the first tool realized by researchers. As mentioned in the previous chapter, an asynchronous collaboration mechanism has been provided through an intuitive graphical hand-attached menu. Such a feature was designed keeping in mind these requirements as well.

Chapter 4

Technologies

In this chapter, the technologies chosen to design the application, both hardware and software, will be discussed.

4.1 Software

The project has been developed as a classical web application.
The server has been realized with Node.js exploiting the Http, Https, Express, and Socket.io modules. Meanwhile, the client-side has been based on the A-Frame WebVR framework, along with JavaScript libraries such as D3.js and the WebSpeech API.
The code for web pages has been written in HTML, CSS, and JavaScript, whereas the server code is entirely in JavaScript.

4.1.1 Node.js

Node.js[1] is an open-source server environment that can run on several platforms (e.g. Windows, Linux, Unix, Mac OS X, etc.), allowing to use JavaScript on a server.
It runs single-threaded, non-blocking, asynchronously and it is event-driven. This means that upon each connection a callback will be executed, but when there is nothing to be done Node.js sleeps. Indeed, the multi-threading model has proven to be inefficient and difficult when applied to networking. Since

[1]https://nodejs.org/

there are no locks, it also prevents deadlocks. Moreover, no functions in Node.js perform directly I/O, so the main process never blocks, bringing several benefits to scalability. It can accomplish any traditional task of a server, such as file handling, dynamic page generation, form data collecting and database management.
Just like JavaScript libraries, Node.js supports the integration of different modules to implement several functionalities. Many modules are built-in and do not require any further installation, while others can be downloaded using the Node Packet Manager (*npm*). In particular, besides the HTTP and HTTPS built-in modules, two additional packages were installed: Express and Socket.io.

Express

Express[2] is a Node.js framework specific for web applications and has been used to quickly set up a web server. It makes very easy to develop an application that can respond to various HTTP requests.
In particular, the framework is based on handler functions that define what has to be executed when a client sends a command to a specific path. This way to direct the client to the different parts of the application is called *routing*.

Socket.io

Socket.io[3] is a technology that allows real-time bidirectional event-based communication between web pages and the server easily and reliably. Based on web sockets, it is supported by any browser or device and is often used by many companies to implement instant messaging chats and enable concurrent changes on a document.
It is made of two components: a Node module running on the server (the actual *socket.io*) and a JavaScript library included in the pages and loaded by the browser (*socket.io-client*). The operation is very simple: an object abstracting the socket is created on both sides and, after that, custom events can be listened to or emitted in each direction to exchange any kind of data, which may be encoded as JSON or binary blob. An event can also be emitted from the server in broadcast to all the clients.

[2] https://expressjs.com/

[3] https://socket.io/

The strength of this approach is that everything can be implemented within few lines of code, differently from previous PHP-based web applications.

4.1.2 A-Frame

Motivations

At the end of 2.3 it has been decided to distribute the application across the web. Hence, there are two possible choices: to develop with Unity and then export to WebGL (Web-based Graphics Library) or to develop natively in WebVR, an experimental API that allows an higher level of abstraction than WebGL.
Both approaches offer cross-platform support to the majority of modern headsets and desktop-based environments. Besides, while the most complete WebVR frameworks are at their first versions, Unity is more than a decade old and has a large asset store. Unity is based on C#, which is synchronous and strongly typed, while WebVR is based on WebGL and JavaScript, which is loosely typed and asynchronous, leading to possible unexpected issues when loading contents on the scene. It is also difficult to find WebVR add-ons that are stable through different versions and support among browsers is diversified.
On the other hand, Unity involves long build times and it may be troublesome to switch huge executable files among clients. Due to hardware limitations, moreover, it is not obvious that Unity can run fluidly on any machine. Furthermore, learning from scratch the fundamentals of a game engine can be time-consuming and, initially, it could be a serious difficulty to bring ideas to life. Plus, Unity requires an HMD to test VR content.
WebVR, even if it is a younger technology and has not been standardized yet, leads to a higher speed in development, allowing to produce a proof of concept in a very short time. This can be achieved by using just a few lines of HTML and JavaScript, with which most programmers are already familiar. Moreover, several templates and tutorials are available to speed up the development process, while complex functionalities can be integrated by exploiting ready-made components. Since it is based on web pages, WebVR does not need to be compiled and distributed. It is not even necessary to install any additional software since any editor suited for web programming can be used to write the code. Such applications can be accessed easily and immediately

from multiple clients just by using a web browser and multi-platform support does not imply re-build. Furthermore, testing does not require to have an HMD, except for specific features (e.g. controller interactions), considering that the program can be explored also from a desktop or mobile setup.

Framework

Several WebVR frameworks were available at the time of this research. The A-Frame[4] framework, in particular, has been chosen over the others (e.g. ReactVR, JanusVR, PrimroseVR) since it provided an higher level of abstraction, enabling to create complex scenes even having a limited experience with JavaScript and no previous knowledge of the three.js library, on which all the mentioned frameworks are based. Besides, the A-Frame community is one of the largest and most active concerning VR: both creators and expert developers often engage to solve programming issues, frequently adding new components to the library.
Originally developed within the Mozilla VR team, A-Frame is now maintained by developers from Google and Supermedium. It currently supports the majority of VR headsets such as HTC Vive, Oculus Rift, Windows Mixed Reality, Google Daydream, Samsung GearVR, Google Cardboard, and Oculus Go, allowing also to use the related controllers. The framework has proven to be powerful and reliable, so it has been used by several companies like Google, Microsoft, Disney, Samsung, CERN, NASA, which also brought important contributions.
It is based on declarative HTML and an entity-component-system (ECS) architecture. It is also optimized for performances: 3D objects are updated in main memory with a little overhead, leading to a good frame rate. A visual inspector of the scene (see Figure 4.1) can also be opened into the browser through a key combination. To enable A-Frame one just needs to add its main script inside an HTML page and to use the *a-scene* tag.

The ECS pattern is also used by Unity and it has proven to be very suitable for VR development. It also leads to great flexibility, lower complexity, and reusability of code. Entities are container objects to which components can be attached. Every object in the scene is internally an entity, but without components, entities neither do nor render anything, similarly to empty HTML

[4]https://aframe.io/docs/0.9.0/introduction/

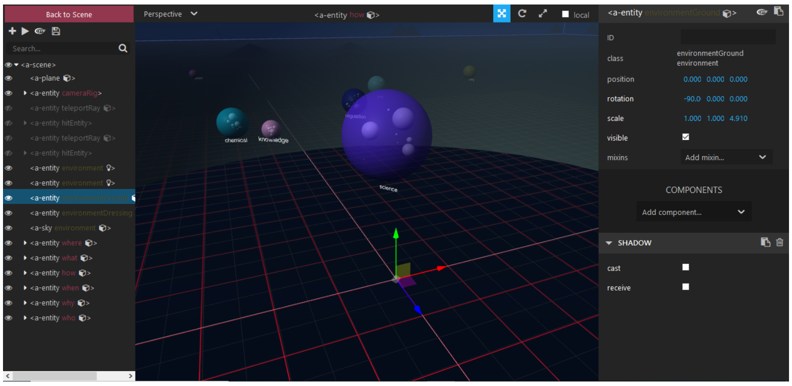


Figure 4.1: A screenshot of the A-Frame built-in inspector

divs. Components are reusable code modules that can be attached to entities to provide appearance, behavior, and functionalities. They can also contain data and receive parameters. All logic is implemented through components, and new types of objects can be defined by mixing and configuring components. The *registration* (i.e. the creation) of new components is fundamental to provide brand new features. Systems are used to carry out global-scope actions on classes of components. They should handle logic, while components should handle data, but are rarely used since, as mentioned, components can easily do both. Specifically, entities are represented inside the *a-scene* using the *a-entity* tag. Components are essentially HTML attributes of an *a-entity*; they are created as JSON objects containing a data schema and specific methods that are automatically called during their life-cycle: *init* will be executed when the component is attached, *update* when its data is updated, *remove* when the component is detached, and *tick* on each scene frame change. This approach fits very well with an asynchronous language like JavaScript. Not necessarily, indeed, an entity or an attribute that has been appended to the HTML tree can be immediately accessed. Lastly, systems are represented as *a-scene*'s HTML attributes. Objects' basic properties like geometry, position, light, and material are abstracted by corresponding components provided by the basic A-Frame library.

As mentioned, A-Frame is based on three.js, but this underlying API can still be accessed through the framework. Specifically, an *a-scene* element is based on a three.js scene and, similarly, an *a-entity* is mapped to one or more three.js objects. When two entities are nested in the HTML tree, so will be their respective 3D objects. Besides, the child's position, scale, and rotation are to

be considered as expressed from the parent's reference point. However, local coordinates can be converted to global ones, and vice versa, by the means of matrix transformations. Such transformations are exposed by the three.js library as simple functions. Anyway, HTML nesting is an easy way to bind together two objects.
In addition, A-Frame allows the DOM (Document Object Model) API to interact with the ECS architecture, creating a bridge between traditional web development and VR applications: entities can be selected using CSS selectors, components can listen to or emit events, entities can be dynamically added to or deleted from the HTML tree, just like components can be attached or removed from any entity. Since DOM allows us to query objects based on their HTML attributes, similarly entities can be filtered according to the set of components they own.
Components may use any kind of JavaScript library (e.g. three.js, Web APIs), enabling an high level of extensibility. Each component does not need to know about the others and can be tested in isolation. Components can be easily shared among developers and exploited just by including the JS script in which they are registered and plugging them to an entity via an HTML attribute. In this manner, an expert developer can build complex components (e.g. physics system, support to controllers, graphic features) that novices can use in HTML without even touching the related JS code. Hence, if an application is built entirely using components, the whole code becomes modular and can be reused in other projects.
Moreover, a set of components that are usually used together can be grouped in a declarative way inside HTML tags with unique identifiers, called *mixins*. These tags, then, must be listed before the *a-scene*, within an additional specific tag called *a-assets*. *Mixins* are particularly useful when, through their IDs, are attached to dynamically generated *a-entities*. In this way, such entities will obtain all the components contained in the *mixin* with a single JavaScript instruction. Since *mixins* are declarative, they improve the readability of code, and can be easily modified by changing the single portion in which they are defined.

4.1.3 D3.js

Data-Driven Documents (D3)[5] is a JavaScript library that has been used to easily bind data, i.e. the semantic model, to HTML entities, specifically *a-entities* introduced by A-Frame, and build complex scenes with few instructions.
In particular, the library is based on the concept of *selections*, still defined using CSS selectors, to abstract a set of nodes in the DOM. Complex actions can be executed on selections (e.g. add event listeners, set attributes, append child nodes, sort) at once in a declarative way, without having to use verbose statements or manual iterations. As stated, data in different formats can be bound to selections with a one-to-one mapping and the properties of DOM objects can be built from such data simply using callbacks.
D3 merely relies on the browser's APIs, but expose them in an easier way to use, introducing a little overhead to allow great graphical complexity at high frame rates.

4.1.4 WebSpeech API

This API[6] developed by Google and Mozilla was used to enable speech recognition within the application.
Nevertheless, it is currently compatible only with Google Chrome and Chrome for Android, which is limiting and inconvenient, but is totally free and has a very simple interface. To overcome the problem of compatibility, it was initially considered also the Cloud Speech-To-Text API[7] which, however, is chargeable and exposes only a limited version for free. Because of this, the WebSpeech API has been selected and the issue of restricted browser use has been resolved as described in Section 5.1.3.
Specifically, the API consists of two components: *SpeechRecognition* and *SpeechSynthesis*, but only the first one has been involved in this thesis project. Essentially, it is a JavaScript library that receives audio from the microphone, exploits a web service for speech recognition to check the audio against a grammar, i.e. the vocabulary allowed in the application, and, when a word or phrase is recognized, returns it as a string.

[5] https://d3js.org/

[6] https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API

[7] https://cloud.google.com/speech-to-text/

The web developer does not need to worry about the speech recognition process: what needs to be done is to include the library script, declare the grammar as an array of terms and instantiate an object which represents the control interface. This object has methods to start/stop the recognition and several callbacks can be defined to be invoked when recognition or an error occurred. Considering that, as mentioned, the speech is sent to a web service, the API does not work offline. Privacy of the users is taken into account, so the permission to use the microphone must be explicitly granted and the pages must be provided using the HTTPS protocol.

4.1.5 Hosting

In order to make the application accessible anytime from anywhere, it has been uploaded on a virtual web server owned by the DEI group at University *Carlos III* of Madrid, where it is still currently hosted and reachable via URL[8]. In particular, the *forever* Node module[9] has been installed on the machine to allow the server script to run continuously.

4.1.6 Atom

All the code has been developed using Atom text editor[10] since it is really light and versatile for web developing. It allows several shortcuts, autocompletion and an effective overview on the code.

4.2 Hardware

In the following are described the hardware devices used to develop the present work.
The main application can be executed from any PC using any modern browser, even though, concerning WebVR, each headset works best with specific browsers[11]. Specifically, Oculus Rift and Oculus Touch controls, provided by the DEI team, were used to develop and test the VR experience. This headset achieves the

[8]http://dei.inf.uc3m.es/bubblevr/

[9]https://www.npmjs.com/package/forever

[10]https://atom.io/

[11]https://webvr.info

best performance when used with Firefox or Supermedium on Windows. Anyway, A-Frame guarantees portability to several modern headsets[12].

4.2.1 Oculus Rift

Oculus Rift CV1[13] (Consumer Version 1), also known as Oculus Rift, is a VR headset released by the Oculus company in March 2016 as the first commercial version after several prototypes. It is one of the most advanced ones and needs to be connected to a computer to operate.

Figure 4.2: Oculus Rift VR headset

It consists of two stereoscopic OLED displays, which provide lower latency than traditional LCD, each one with a 1080×1200 resolution, a 90 Hz update frequency, and a 110° field-of-view. The distance between the lenses can be adjusted through a slider, just like the gap between the helmet and the user's eyes, adapting to different facial shapes, and allowing to wear glasses while using the device. It also incorporates headphones to support spatial sound effects along with a built-in microphone. There is also a proximity sensor that detects when the helmet is being worn.
As can bee seen in Figure 4.2[14], the headset comes with two infrared sensors, similar to lamps that are placed on the user's desk. Subsequently, Oculus

[12]https://aframe.io/docs/0.9.0/introduction/vr-headsets-and-webvr-browsers.html

[13]https://en.wikipedia.org/wiki/Oculus_Rift_CV1

[14]Painter, Lewis. "Oculus Rift." Tech Advisor, IDG, 21 Nov. 2018, www.techadvisor.co.uk/cmsdata/reviews/3643626/oculus_rift_201709.jpg.

Touch have been released, i.e. two tracked controllers featuring an analog stick, three buttons, and two triggers, that bring the user's hands in VR. Rotational and positional tracking are implemented through a mechanism called *constellation*. The helmet and the ring in each controller are equipped with several infrared LEDs which blink according to a specific pattern. By knowing the exact position of the LEDs and their blinking pattern, the sensors can track the devices with an extremely precise accuracy and near-zero latency. In this way, the Oculus Rift enables 6 degrees of freedom, allowing users to stand, walk, sit and even jump around the virtual environment.

Chapter 5

Development

In this chapter, the whole development process will be described, focusing on each feature and the solution provided for it.
At first, the server-side of the application will be discussed, then the behavior of the web pages that implement the actual VR application will be analyzed.

5.1 Server-side

Given the browser-tied nature of the WebVR API, the server has a rather marginal role in the application. All its source code is contained inside a script named *app.js*, which can be run by using the *node* command on the corresponding machine.
Its functionalities simply involve providing access to the pages, managing of the files depicting the semantic model, and handling of vocal commands throughout different browsers. Each one of these features is outlined below.

5.1.1 Providing application

As mentioned in Section 4.1.4, the HTTPS protocol needed to be enforced in order to enable the use of the microphone. Anyway, the secured version offers several advantages over HTTP and does not cause any significant drawback to the application. Thus, an HTTP and an HTTPS server have been set up using the respective modules offered by the Node.js standard package. The first just listens on the standard 80 port and redirects every request to the latter. As requested by the protocol, a TLS certificate for the server was thereby needed. Since the machine on which the website would be hosted was still unknown

at that moment, it was conveniently decided to use a self-signed certificate, obtained through the OpenSSL toolkit, to overcome the issue. Due to this, a warning is prompted by the browser on each connection attempt.
Besides, a web socket is initialized to ensure future bidirectional communications with clients.
Beyond that, when contacted with HTTPS, the webserver just routes the client to its root directory, where the actual pages are stored. As usual, the *index.html* file represents the entry point for the application.

5.1.2 File managing

To allow users to share their views on the semantic model, a mechanism to save and load different versions of the related JSON file has been realized. Using the interface exposed by Socket.io, indeed, the server can be informed of the intention of a client to store a specific taxonomy file or to retrieve the list of the already saved views, which are stored in a dedicated folder in the server file system.
This has been implemented by enabling it to listen to two distinct events: *save file* and *list request*. The callback associated with the first event receives the file and the name chosen for it by the user, and stores it in the folder. Progressive numbers are chained to the file name in case of duplicates, in order to avoid conflicts. The second callback sends back to the client an array containing the list of file names already stored. The client can subsequently load a specific one by reloading the page and including the name in a query parameter.

5.1.3 Vocal commands handling

As anticipated in Section 4.1.4, the adoption of the WebSpeech API involved the severe limitation of capturing microphone audio only by using the Google Chrome browser.
As can be seen from the architectural schema depicted in Figure 5.1, a solution based on two web pages has been implemented.
By leveraging the established bidirectional socket channel, the server can exchange recognized text between the page where the audio is being recorded, called *audio.html* and running in Chrome, and the one in which the main application is being executed, i.e. the above-mentioned *index.html*, opened into another browser (e.g. Firefox). This is done with no significant perceived delay from users.

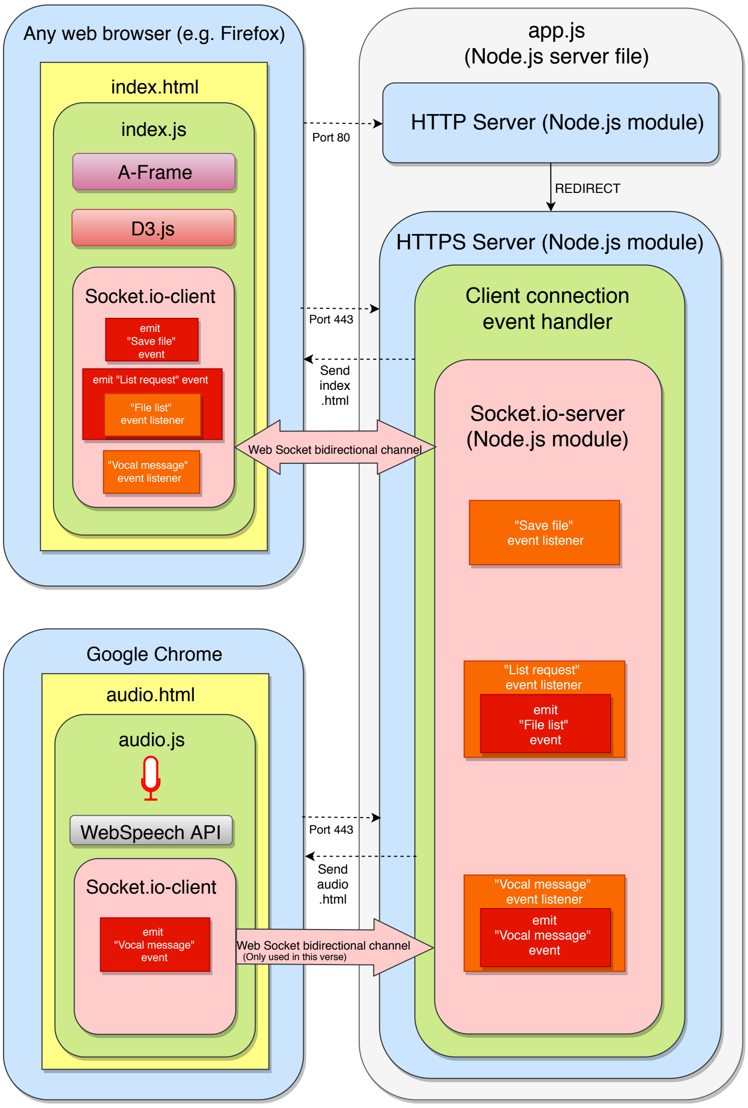


Figure 5.1: Architectural schema of the designed web application

5.2 Client-side

The core of the application runs on the client-side. As mentioned in the previous Section, two pages have been used to implement all the functionalities of the project: *index.html* and *audio.html*.
The main page has been formatted according to the A-Frame guidelines[1], and

[1]https://aframe.io/docs/0.9.0/introduction/best-practices.html

represents the static and declarative interface of the VR scene. The second, instead, is merely a dummy page used to provide visual feedback of the captured audio. It is intended to be opened in the background and then may also be disregarded, since, as will be seen, the recognized speech is shown in the virtual environment as well.
Each of the two pages includes a related JS script file. In the first case, it contains the functions and components concerning the application logic, while the latter consists of the commands and callbacks that exploit the WebSpeechAPI.
A conceptual schema of the *index.html* page is depicted in Figure 5.2. The most relevant components have also been highlighted.

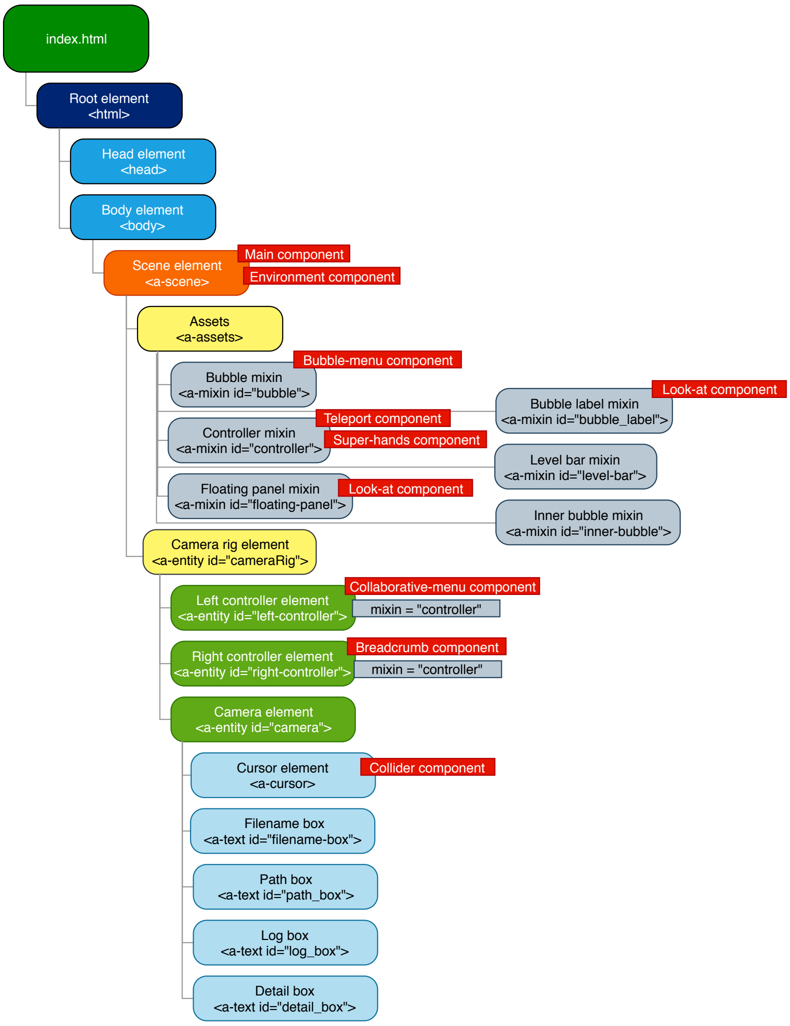


Figure 5.2: Tree diagram describing the DOM structure of the *index.html* page

It is important to remember, as explained in Section 4.1.2, that:

- The main function associated with a certain component is executed only after that the corresponding entity has been loaded on the scene.
- Two nested elements will be bound in the virtual environment.
- Mixins are sets of components. Through their ID, they can be used to easily attach all those components to an entity.

A summary of the main component flow is shown in Figure 5.3. Naturally, as in any GUI application, the main operations will be performed asynchronously. One of the declared listeners, indeed, will be invoked each time the corresponding event is triggered by the user's actions.

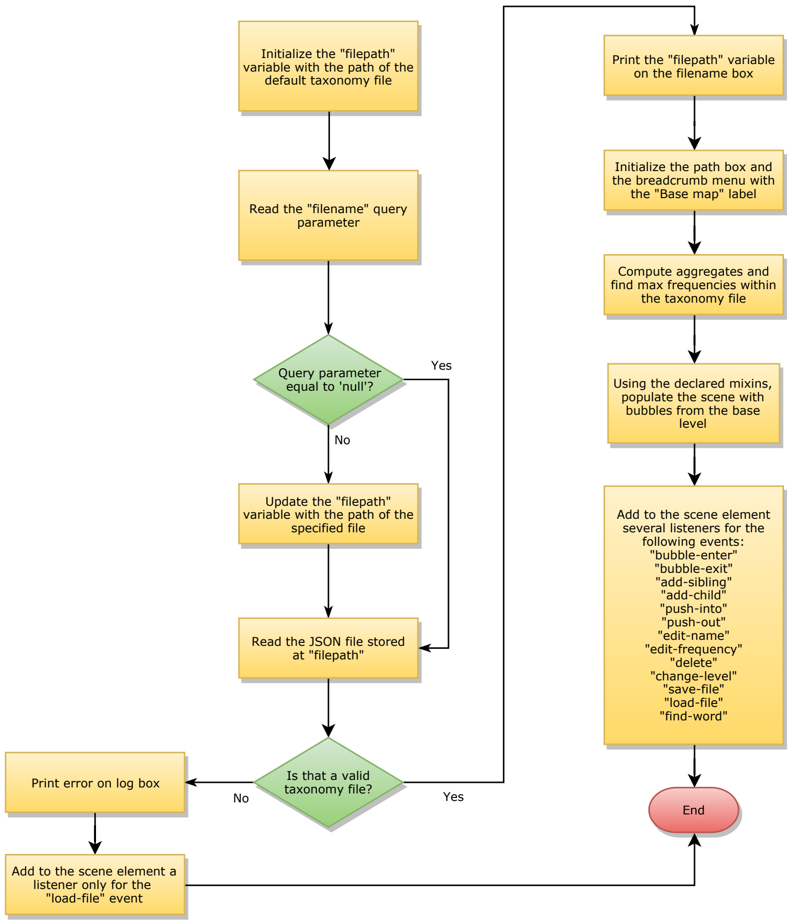


Figure 5.3: Workflow of the application main component

From now on, the design of the implemented functionalities will be described. Focus will be on the problems encountered and on the conceptual workflow provided for them, rather than on the actual application code.

5.2.1 Taxonomy file format

A necessary preliminary step is to describe the format in which the underlying semantic model has been provided. A sample of the JSON taxonomy file obtained through the tool described in [46] is illustrated below:


```
{
  ”where”:{
    ”locations”: {
                ”color”: ”f283a5”,
                ”name”: ”locations”,
                ”terms”: [
                        {
                                ”doubt”: 0,
                                ”freq”: 8,
                                ”name”: ”germany”,
                                ”weight”: 1.5
                        },
                        {
                                ...
                        }
                ]
        }
  },
  ”what”:{...},
  ”how”:{...},
  ”when”:{...},
  ”why”:{...},
  ”who”:{...}
}
```


It consists of nested categories that contain, on the deepest level, an array of terms. Such terms are characterized by several attributes, but for the aims of this application, only the frequency inside the corpus of documents has been taken into account. During development, no assumptions have been made on the maximum nesting depth of a taxonomy, neither on its specific domain. Remarkably, the outermost level has been conceived to foster a journalistic

approach to the exploration, by using the ”Six Ws”.
Such a file is in a folder on the server storage and is read just once by the main component using the D3.js interface. Then, it remains in the main memory and is used as a basis to build the whole visualization. Each correction of the model made through the application is propagated to the corresponding JSON object, so that it remains consistent.
As will be discussed in Section 5.2.7, changes can be permanently saved back on the server storage, but either way, the same format will be adopted.

5.2.2 Scene design

First of all, it has been decided to build an environment that stimulated the user's senses and minimized distractions. Surroundings needed to be minimal, but nevertheless immersive. To create such a background without designing it from scratch, the environment component[2] from the official A-Frame Registry has been exploited. The *tron* setup, in particular, has been selected, since it was the most basic and did not affect frame rate.

Text boxes

Four text boxes have been placed right in front of the camera, providing visual feedback to the user:

- *Filename* box: containing the name of the JSON file associated with the current session. By default, *New Project* will be shown.
- *Path* box: containing an ordered list of the explored levels. It helps users to remember their position within the model.
- *Log* box: it will hold instructions or report errors, to guide users during exploration.
- *Detail* box: used with the dual purpose of providing details on bubbles when hovered and displaying the received vocal commands on the scene.

The position of the mentioned boxes inside the scene is shown in Figure 5.4.

[2]https://github.com/supermedium/aframe-environment-component

Figure 5.4: The four text boxes attached to camera

Bubble design

The idea of using bubbles turned out to be still adequate to provide a simple, but yet effective, representation of categories and terms. Floating text labels under each bubble, besides, have been used in both cases to picture their names. Specifically, a category, at any level, has been represented as a colored semi-transparent sphere. Moreover, if a category is not empty, the related sphere is filled with smaller opaque orbs. On the other hand, individual terms have all the same radius (set to 1 unit, i.e. meters) and looks exactly like an empty category, but with a semi arc on the top, a sort of progress bar.
Such representation, moreover, should help users to gain insights on the dataset. That's why the level of each term progress bar and the size of the category spheres have been computed according to their importance within the taxonomy. In particular, the most frequent term inside the corpus will have a full bar. The bar filling percentage (*BFP*) for the other terms, instead, is determined according to the formula:

$$BFP = \frac{freq_{term}}{freq_{maxOverall}} \tag{5.1}$$

Such a bar has been further colorized in red, yellow or green to denote that it is under 33%, over 33%, and over 66%, respectively (see Figure 5.5). Colors have been used to deliver an even more immediate overview.
Meanwhile, a category's radius will be greater as more important are the terms inside it. Its impact is quantified by aggregating the frequencies of such

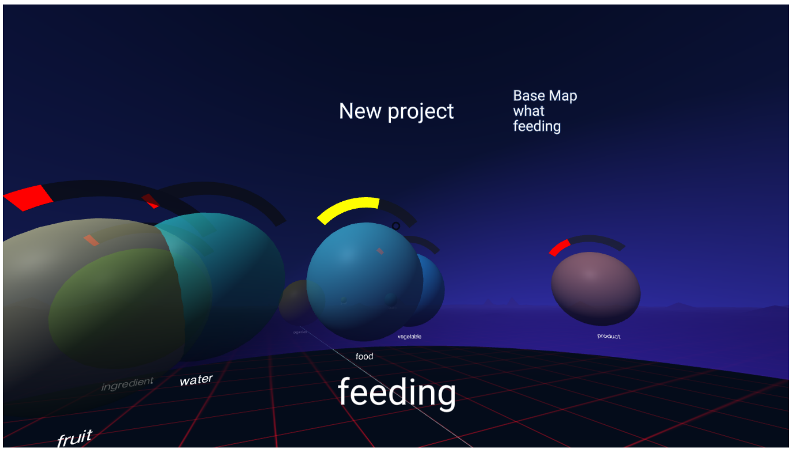


Figure 5.5: A group of terms inside the *feeding* category

contained terms. This new extent has been named *freqSum*. Iteratively, if a category contains other categories, its influence will be computed by summing up their *freqSum* too. This approach is also compatible with categories containing both individual terms and other categories at once, even if this is not expected to happen in a valid semantic model. Furthermore, the size of each category has been normalized to the level in which it stands. This means that, at any depth, there will be a category with a maximum radius, i.e. the most important in that level, and the radius of the others will be obtained in respect to it:

$$R_{category} = \frac{freqSum_{category}}{freqSum_{maxInLvl}} \cdot R_{max} \tag{5.2}$$

R_{max}, as for the term radius, has been set to 1 unit.
An identical approach, to highlight proportions, was applied also to the smaller inner bubbles inside each category (see Figure 5.6). The maximum radius that an inner bubble may have, $R_{innerMax}$, is expressed as a function of the radius of its category:

$$R_{innerMax} = \frac{R_{category}}{4} \tag{5.3}$$

And, similarly, for each inner bubble, its radius R_{inner} will be:

$$R_{inner} = \frac{freqSum_{innerElement}}{freqSum_{maxInCategory}} \cdot R_{innerMax} \tag{5.4}$$

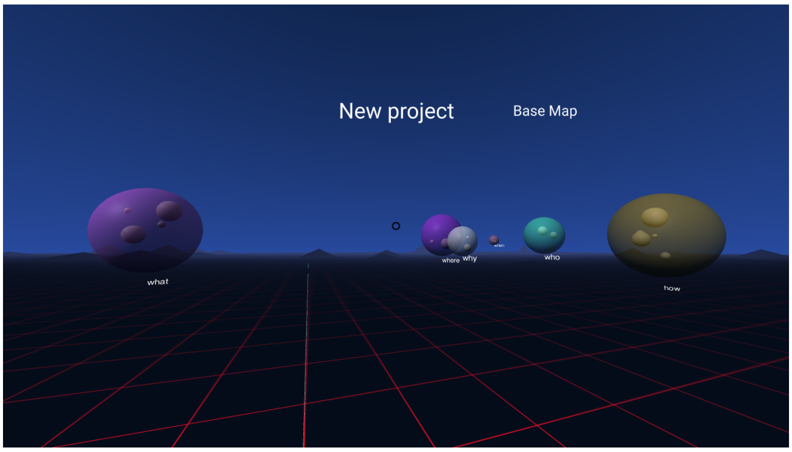
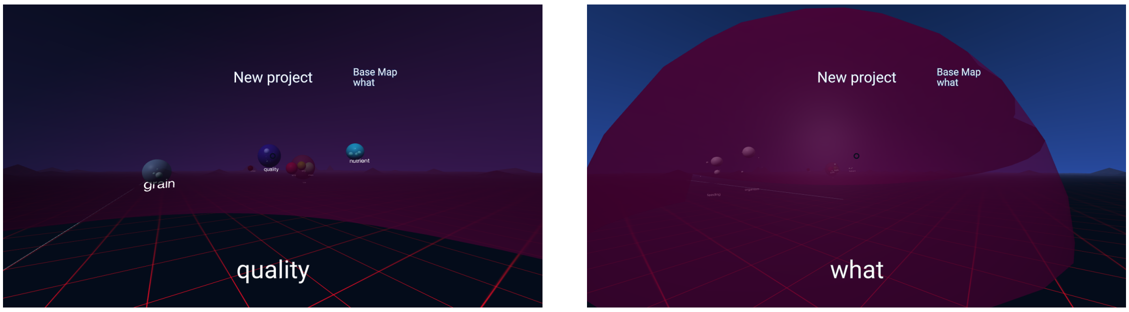


Figure 5.6: The six main categories as they appear in the base level

Such operations required a recursive function that, iterating over the taxonomy file, finds the maximum values and computes aggregates. This function is called before the bubbles for the current level are created.
The color of the bubbles is read from the taxonomy file, if available, or it is chosen at random.
Due to perspective distortion, the different sizes of the spheres might be incorrectly perceived by users. As suggested in [40], then, it has been decided to use a common feature among all entities. The text label representing the bubble name, indeed, has the same size for each element. The *look-at* component[3] has been attached to these labels, so that they follow the camera, and can be read from any angle.
Furthermore, when the users are inside a bubble, this latter one surrounds them like a dome (see Figure 5.7). It is also semi-transparent and allows them to see the elements in the previous level. In particular, to avoid confusion, bubbles outside the dome will be opaque, while those on the inside will retain their see-through appearance. This feature has been implemented because it was explicitly suggested during the previous exploratory study [47].
To ensure generality, the look of each level is built repeating the above-mentioned steps. In this way, users can move back and forth through the nesting model (see Section 5.2.4), until they enter a term bubble. In that case, they will not be able to go deeper, and the scene will not contain spheres anymore (see Section

[3] https://supermedium.com/superframe/components/look-at/

5.2.5).



(a) View from the inside (b) View from the outside

Figure 5.7: The *what* main category shown as a dome after entering it

Graphical menus

As mentioned in Section 2.3, graphical menus have been used, enabling users to perform complex tasks. The A-frame GUI component[4] was employed to provide buttons and icons. Such menus were designed to be as more intuitive as possible and had to be shown/hidden on-demand.
Several works like [48] and [49], pointed out that, in a VR environment, 3D buttons may lead to shorter execution times and higher precision than flat 2D ones. Specifically, two menu layouts mainly emerged: a pie (i.e. circular) menu and the more traditional vertical list. The former is more adequate for contextual menus with many options, since the radius can be varied to keep it compact; in particular, this layout paired with ray-casting controls has resulted in a nearly zero error rate. The latter, instead, is faster when few options have to be displayed in a world-fixed position.
Based on that, three kinds of menus were realized:

- A circular menu attached to each term or category. It allows users to perform all kinds of correcting actions on the semantic model (see Section 5.2.6).
- A vertical menu attached to the left wrist used to manage multiple taxonomy files (see Section 5.2.7).

[4]https://github.com/rdub80/aframe-gui

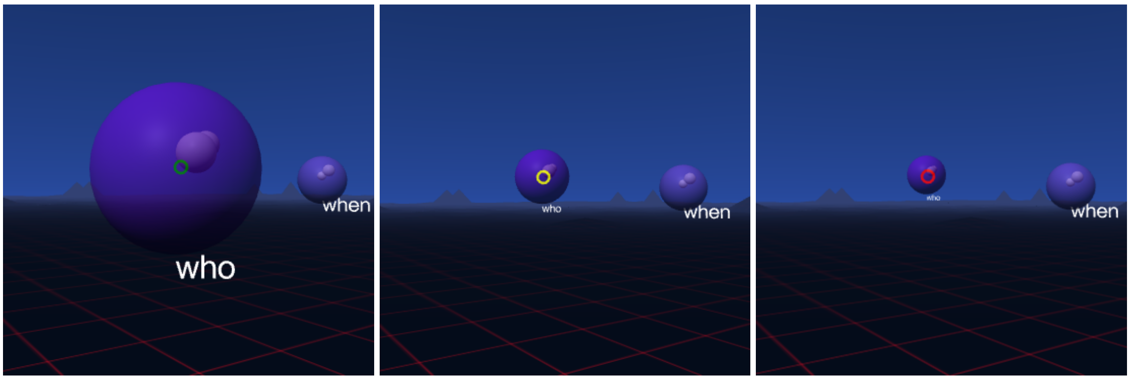
- A vertical menu attached to the right wrist, serving both as a breadcrumb and as a way to move quickly across non-adjacent levels (see Section 5.2.4).

5.2.3 Interaction

Multiple ways of interacting with the environment were implemented inside the visualization, both in the immersive and the desktop setup.

Basic interactions

Even without an HMD, A-Frame offers traditional mouse and keyboard support, by default. Under these conditions, users can use WASD keys to move, while they can click and drag the mouse pointer to look around. A circular *cursor*, placed right in front of the camera, can be used to make selections within menus. Additionally, when the cursor hovers a bubble, its name is shown inside the *detail* box mentioned above. Such a mechanism, inspired by [32], has been provided to allow users to check out the surroundings without even moving.
Bubbles can be also moved around with the cursor. To enable this feature, a custom interaction mechanic has been implemented. The way it functions can be observed in the Figure 5.8 here below.



(a) Attracting (b) Repulsing (c) Re-positioning

Figure 5.8: The three-stage mechanism to move bubbles through the cursor

When an user clicks on a bubble for 1 second, indeed, the cursor will turn green and the corresponding bubble will be automatically attracted to the

user's current position, only a step away (a). If the user keeps clicking for one more second, instead, the cursor will turn yellow and the bubble will be pushed further, but still within reach (b). Finally, when users keep clicking on the same bubble for 3 seconds, the cursor will turn red and the position of the sphere will be restored as it was before they started clicking (c).
In this basic setup, since hands are not tracked, wrist-attached menus will float in a fixed position. Their visibility can be toggled through the Q and E keys. Similarly, a contextual menu can be shown, or hidden, for each bubble by clicking on the corresponding sphere.
Even if limited, these cursor-based interactions, still allow users to take full advantage of each functionality provided by the application.

Gaze

Gaze-based interaction, useful both for 3DOF and 6DOF systems, has been included too. When wearing an HMD, indeed, the above-mentioned cursor can be moved by simply rotating the head. This is the only way to operate when using headsets that provide merely rotational tracking and single-button controllers (e.g. Samsung GearVR, Google Daydream, Oculus Go).

Tracked controllers

More interesting, however, is the interaction through tracked controllers, which allows to use both arms and hands, making the VR experience truly immersive.
A-Frame exposes several components to include this type of input within the scene[5]. Specifically, support to major 6DOF tracked gamepads (Oculus Touch, HTC Vive, Windows Motion) has been implemented.
Controllers are automatically detected and rendered into the environment through their 3D model (see Figure 5.9). A light blue laser beam is shot out each one of them and, both button pressings and intersection events can be used to encode several actions.

Laser controls offer all the functionalities of the cursor, but with an aim-and-shoot approach, in a more natural and engaging way. Furthermore, the *super-hands*[6] component has been used to easily translate specific gestures

[5] https://aframe.io/docs/0.9.0/introduction/interactions-and-controllers.html

[6] https://github.com/wmurphyrd/aframe-super-hands-component

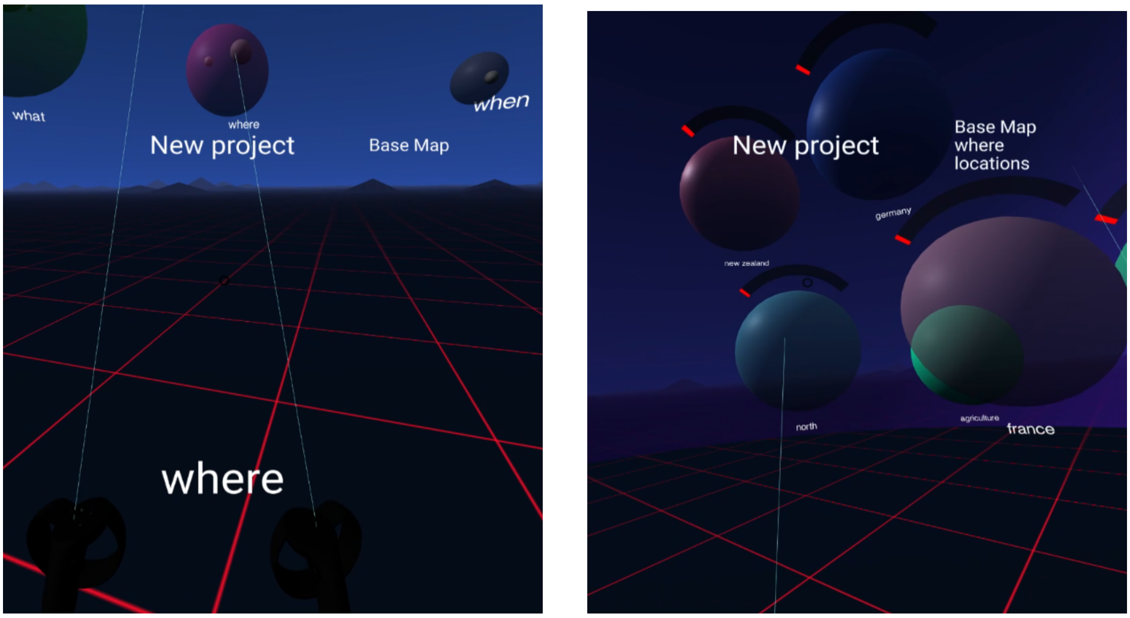


Figure 5.9: The laser beam shooting controllers portrayed within the VR environment

into actions performed on the entities. Thus, bubbles can be grabbed and moved around, holding down the *trigger* button while intersecting. Bubbles can also be brought closer, or further, pressing one of the two controller main buttons while they are being grabbed.
Finally, the two hand-attached menus can be shown on-demand by touching the corresponding *thumbstick*.

Vocal input

As mentioned in Section 5.1.3, audio is captured through the *audio.html* page and its related script, *audio.js*. This script, exploiting the WebSpeech API, can recognize the user's words. The page interface is very basic (see Figure 5.10.a). Plus, it offers a quick way to test the feature: by saying one of the depicted color, the page background will change according to it.
After the user's consent, audio is being constantly recorded through the microphone, and results are then delivered to the application page via socket event. There, a function filters out reported errors and trim the received string (e.g. removing uppercase letters). The vocal input is always shown on the scene by placing it in the *detail* box, at the bottom of the screen (see Figure

5.10.b).

(a) The *audio.html* page (b) The recognized speech is shown in the VR application

Figure 5.10: An example of how vocal commands work

Received speech is consequently used to carry out one of the following tasks:

- *Insert text strings*: any time a textual input is needed (i.e. to edit a bubble's attributes or to specify a new file name), the received string will be written on the corresponding menu button. This is essential in the immersive setup wherein, due to the worn HMD, the keyboard is not accessible.
- *Attract bubbles*: when a certain bubble is visible on the scene, it can be moved, through an animation, near to the user's current position just by saying its name. This feature is useful to quickly move entities around, especially when tracked controllers are not available.
- *Search for a term or a category*: on the other hand, if a bubble is not currently shown on the scene, it can be searched by saying its name. A *find-word* event will be sent[7] to the main component, which solely can access

[7] Actually, events are not emitted on components, but on entities. In fact, the event is emitted on the *a-scene* element. Since *a-scene* event listeners are placed only inside the main component, though, from now on, this will be taken for granted and it will not be specified anymore. This difference is not so relevant to understand the application workflow.

the taxonomy object. If the word is present within the model, the absolute path to reach it will be shown inside the *log* text box.

- *Exit a bubble*: when exploring an inner level, users can exit from the current bubble, and instantly return to the previous level, just by pronouncing the word *"back"*. A blink animation has been used to ease the transition. Implementation details are explained at the end of Section 5.2.4.

5.2.4 Navigation

A crucial aspect of the model exploration is played by movement, both through the current scene and between nested layers.

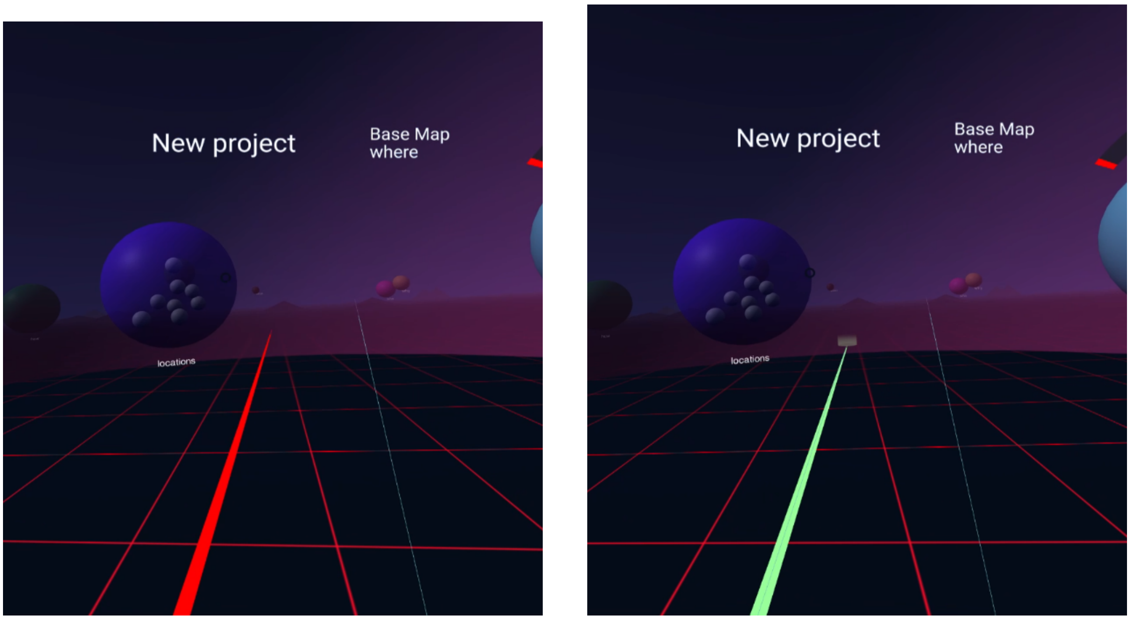
Teleport controls

As said, WASD keys can be used to intuitively move across the environment in a non-immersive setup. On the other hand, since keys were precluded, another solution had to be designed for the HMD case.
Hence, the *teleport-controls* component[8] has been attached to the hand controller entities to enable an alternative navigation method. Specifically, when the *grip* button is held down on one of the controllers, a thicker laser ray will be cast out. When the ray is pointed to an invalid position for users to occupy, it will be painted in red (see Figure 5.11.a) and teleportation will not be allowed. If the laser intersects the ground in a valid spot, instead, the ray will turn green (see Figure 5.11.b) and a circle will show the future position of the user. At this point, if the user releases the *grip*, the camera position will be instantly shifted to that place. This option enables to explore even a wide environment while comfortably seated.

Navigating nested levels

Concerning the travel among different levels, an intuitive and straightforward workflow has been designed. The basic concept was to allow users to enter a semantic element by simply stepping into the corresponding bubble. Then, a new scene had to be built. The environment had to significantly change and, furthermore, the camera had to be repositioned at the origin of

[8]https://github.com/fernandojsg/aframe-teleport-controls



(a) An attempt to teleport on an invalid location

(b) A valid example of teleportation

Figure 5.11: Teleport controls are used to move through the VR environment

the scene. That is why a transition animation had to be added to prevent discomfort from users. As mentioned, however, the previous level will still be somehow visible through a large translucent dome, abstracting the bubble that has just been entered. Similarly, therefore, when inside a certain term or category, users might go back to the previous view by walking out the surrounding dome.
This mechanism has been implemented by means of a *collider* component[9]. The collider has been attached to the camera cursor and, at the same time, all the bubbles have been labeled as *collidable* simply through an HTML class. A further custom component, later called *collision-detector*, has been written and attached to each bubble to detect collisions with the cursor. When this happens, a black image is placed just in front of the camera (to simulate a blink transition) and a *bubble-enter* event is emitted to the main component. This latter, reading the taxonomy object, will build the new level scene, will save useful information in a stack structure, and, finally, will remove the black

[9]https://www.npmjs.com/package/aframe-aabb-collider-component

image from the camera. Analogously, when an user steps out of the corresponding dome (and so does the cursor), the *collision-detector* component will display again the black image and signal the main one through the *bubble-exit* event. By exploiting the mentioned stack, the previous level will be restored and the black background will be then removed once again.
As a consequence, bubbles can be entered also standing still and bringing them closer enough to trigger the collision event.
The stack structure is crucial to keep track of the explored views and allows to move back and forth through the model. It is also used to update both the *path* box and the breadcrumb menu.

Interactive breadcrumb menu

Breadcrumb menu was designed to be interactive. As can be seen in Figure 5.12, it will dynamically contain a button for each level in the user's current path.

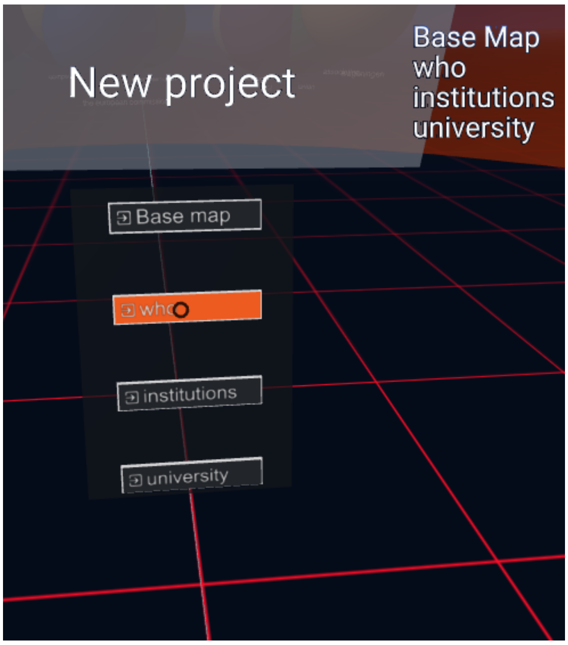


Figure 5.12: The breadcrumb menu shown from the interior of a term

By selecting a button, the user will be instantly brought back to the corresponding level, allowing to quickly jump between non-adjacent layers of the taxonomy. Inside the button handler, indeed, the usual black background will be put on camera and a *change-level* event, containing the desired destination, will be emitted to the main component. Here, the stack will be resized as it would be in the level immediately following the requested one and, then, a *bubble-exit* event will be emitted. This solution has been implemented to partially reuse the previously described workflow. Naturally, If users select the lowest button, no navigation event will be triggered. Besides, a warning will be printed on the *log* box, reminding them that they are already in that level.

Traveling back with voice and cursor

As mentioned at the end of Section 5.2.3, also voice (i.e. by saying *”back”*) can be used to go up a level in the model hierarchy.
Similarly, also the camera cursor can be used to exit from a bubble. When inside a bubble, if users hover the surrounding dome and hold the click for 3 seconds, turning the cursor to blue, they will be brought back with the usual blink animation. This last feature can be really useful in a basic setup, given that in some cases it can be quicker than walking out the bubble dome.
Still, to reuse code, these two options have been implemented in the same way, by emitting a particular *change-level* event toward the main component. Inside this event detail, the name of the desired destination will signal to the component that it just has to bring the user to the previous level.

5.2.5 Term details

Each term in the taxonomy may be accompanied by additional information, useful to provide further details, such as corpus documents regarding it, and related multimedia. Such details can enhance the effectiveness of the visualization with regard to learning about the specific domain. Plus, pictures and descriptions can deliver to users a quick impression of a word, speeding up the data exploration process. Further, a more detailed experience could lead to an increase in engagement.
These related data could not be neglected in the design phase and had to be represented somehow.

CSV catalog

Data are linked to the corresponding terms through a catalog file containing comma-separated values (CSV). Its header is shown here below:

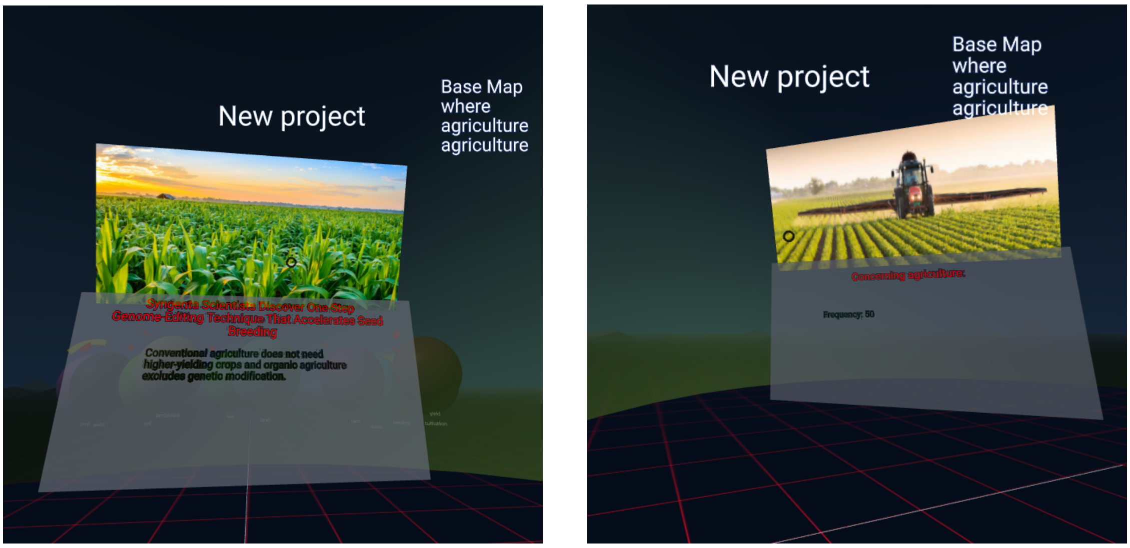
(index, term, media_type, file_URL, description)

As one can see, it is pretty basic. For each line, it just stores the mapping between the term and the associated file, including also its media type and an additional description, which can be empty.
For the realized prototype, only text and images have been taken into account. Articles, documents, and descriptions were gathered from the corpus using the taxonomy generation tool described in [46]. They were then stored in a folder on the server, and their paths written into the catalog . Pictures, instead, were automatically found and collected from Google Images using a Python script[10].
The catalog schema, however, has been kept general enough to support future extensions (e.g. sounds, videos).
Following the *details-on-demand* philosophy, when an user steps into a term bubble, the catalog file is read from the main component using the D3 library. The lines of the catalog associated to that term are then used to put the additional data on the scene.

Representation

The virtual environment can vividly portray a workplace so, like the real world, is quite natural to populate it with two-dimensional data.
In particular, it has been decided to use full-scale floating panels which are 1 meter high and 2 meters wide, to display both images and texts. As can be seen in Figure 5.13, one panel was used to show all textual documents, while another one was designed to hold pictures. They were placed at the center of the term bubble, so that they are immediately visible when entering.
A semi-transparent white plane was used as background for the text panel. It has been done considering that its central position should not completely preclude the view of the environment behind it. The frequency of the term will be also indicated in this panel. The title of each document is shown in red, while its body is depicted in black. These colors were selected due to the high contrast with the white background.
The picture panel, instead, has been maintained fully opaque to properly highlight its own content. For this, it has been placed right above the text panel, so that it does not overlap to any significant portion of the surrounding dome.

[10] https://github.com/hardikvasa/google-images-download



(a) Frontal view, a brief text is shown below

(b) Lateral view, the term frequency is shown below

Figure 5.13: Floating panels used to portray related details inside a term

The already mentioned *look-at* component has been attached to these two planes, so that they will lean and rotate according to the user's position. In this way, they will still be readable from any angle when moving inside the term bubble.
Furthermore, to let the people browse the information inside each panel, they have been made interactive. By clicking (both with cursor or controllers) anywhere on the right side of the panel, the next content will be shown. By clicking on the left, the previous element will be displayed again. In this manner, users can quickly navigate this sort of gallery inside each term.

5.2.6 Bubble menu

Several adjustments can be made on the underlying semantic model. Specifically, all the features of the designed entities were designed to be editable. It may be useful both to correct the output of the preliminary categorization, and to allow users to extend it, potentially creating brand new layers in the hierarchy.
For this purpose, a contextual menu has been attached to each bubble. When

selecting a bubble, its related menu will be displayed right in front of it, tangent to the spherical surface. It contains several buttons, arranged in a radial layout on a black see-through circular background, as shown in Figure 5.14.

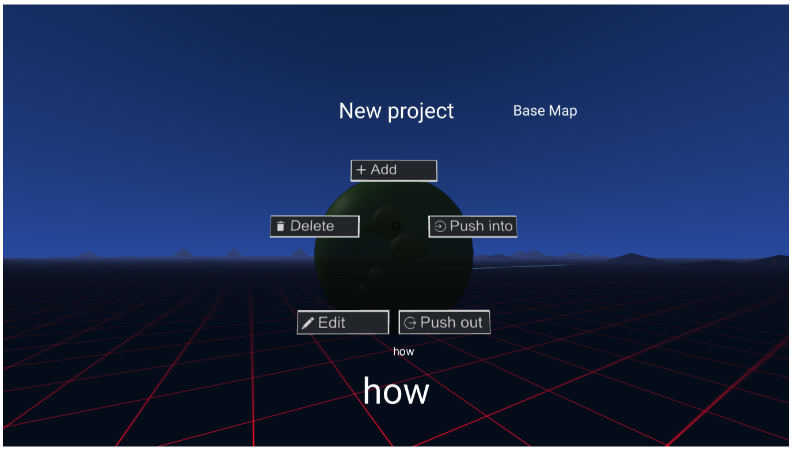


Figure 5.14: The contextual menu attached to each bubble

Once selected, every option can lead to a series of inner menus, which will replace the main one. To go back to a previous menu, users will just have to click anywhere on the circular surface, representing a sort of implicit back button.
Basically, the different button handlers, after performing validity checks, emit homonymous events toward the main component. There, such events are handled by specific listeners that act on the scene and the related taxonomy object.
Each one of these menus allows to execute multiple actions on the specific bubble and, more generally, on the current semantic model. The different commands will be explained below.

Add

This command is useful to generate a new entity on the scene. When clicked, it will first prompt users to insert, using their voice, the name for the new entity. Whenever a word is recognized, it will be put inside the only button, which will display it. The name will then be confirmed by clicking on the button while it holds the desired one. Users will be instructed on how to do this through an advice shown in the *log* box.

Once a name is selected, another menu will allow to specify if the new entity will be a term or a category, and if it has to go inside the specific bubble (child) or in the current level (sibling). Naturally, no entity can be added as the child of a term, since it is already a terminal node in the taxonomy tree.
By default, categories will be created empty (i.e. associated with an empty *terms* array) and terms will have zero frequency. The taxonomy object will be then consistently updated.
If the entity is added inside the selected bubble, a new inner sphere will be instantly generated and will be visible inside it. Similarly, if the word is added on the current level, the corresponding bubble will be placed nearby on the scene and shown right away.

Edit

Through this option, the attributes of the selected bubble can be modified. In particular, categories can only change name, whereas terms can have both their name and frequency changed. A first menu page will ask to users to choose just among that.
Then, the new value for the attribute, either a number or a word, will be provided through speech, with the same exact method described above.
Once the name is changed, the text label under the bubble will be updated. Inside the JSON model, the property corresponding to that bubble will likely change name.
Analogously, when the frequency of a term is adjusted, its level bar will subsequently change filling percentage. Further, the related *freq* attribute inside the JSON taxonomy will be set to that value. For consistency, if the new frequency turns out to be the highest one, the bars and sizes of all the other bubbles will be modified as well.

Delete

As well as new bubbles can be added, existing ones can be deleted. They will be removed both from the scene and the semantic model, along with the whole hierarchical branch below them.
When selected, this option will simply ask to users to confirm or cancel their decision. By confirming, indeed, the selected bubble will just disappear and the corresponding property will be deleted from the JSON taxonomy object. Canceling is completely equivalent to clicking on the menu surface, but has

been explicitly provided due to a symmetry with the other option.

Push into

This option allows to move the selected bubble deeper into the categorization hierarchy. Specifically, both terms and categories can be pushed into another bubble, but this last can only be a category. As mentioned, terms cannot contain other entities.
Two approaches were implemented to select the destination category.
A first possibility, more intuitive, requires users to simply click on the target bubble. Users will be guided from the instructions displayed inside the *log* box. Further, if a term bubble is wrongly selected, a warning will be printed again in the *log* box, reminding that only a category can be chosen. Even it is fast and natural, this method though may often lead to wrong selections when many entities are present on the scene.
Hence, a second selection technique has been provided. Using a more traditional approach, only the names of valid bubbles (i.e. categories), will be shown as buttons in an inner menu, following the alphabetical order. Since the list of entities can grow very fast, such a menu was divided into pages, that can be navigated through clickable arrows. When found, the destination category can be selected by clicking on its name. This technique is definitely slower than the first one, but it enables to make a more accurate selection of the target, useful in a crowded scenario.
Once the target category is chosen, however, a quick animation will show the early selected bubble moving into it, disappearing from the current level. A new inner sphere will also be generated inside the target category. Similarly, the corresponding property will be consistently moved within the taxonomy object.

Push out

In a dual way to the previous option, this one allows to move bubbles to an higher level of the hierarchy, out of the current category. It has no inner menu associated, since the task can be carried out unambiguously.
When this button is clicked, indeed, the selected bubble will move, through a smooth animation, out of the surrounding dome, and will then become opaque. This animation should intuitively suggest to users that the entity does not belong to the current level anymore. Equally, the related property will be

moved inside the hierarchical model.
Naturally, if users choose this option at the initial level of the taxonomy, called *base map*, a message in the *log* box will remind them that they are not inside any category yet.

5.2.7 Support to collaboration

A vertical menu attached to the left wrist has been implemented to support asynchronous collaboration. It enables to resume a previous exploration, or to analyze someone else's results, through an approach based on saving and loading different taxonomy files.
As explained in the previous Section, through this application, the underlying semantic model can be modified in many ways. As specified, though, such adjustments are propagated only to the JSON taxonomy object in main memory. When users want to make these changes permanent, they can do it by means of the above mentioned menu, shown in Figure 5.15.

Figure 5.15: The save & load menu that supports asynchronous collaboration

Similarly to the one designed for bubbles, users can go back to a previous menu page by clicking anywhere on the menu background.

The actions associated to each button are outlined as follows.

Save as...

The current taxonomy object will be stored in a file with a given name. Such a name will be inserted through voice with the usual mechanism already described in Section 5.2.6. In case of conflicts, duplicates are automatically avoided by postponing a progressive integer to the file name.
Through the socket interface, the handler for this button will emit a *save-file* event to the server. The content of the JSON object will be specified inside the event details. From now on, the file will be stored in the *saves* folder on the server storage.
Then, the *filename* box will be updated, displaying the associated file name also within the current scene.

Save

This button works just like the previous one, with a slight difference. When clicking on it, no file name will be requested, and changes will be automatically saved to the current file, that is the one shown inside the *filename* box. If it is selected in a new session, before saving for the first time, it will behave exactly like *Save as* and a name will have to be provided anyway.

Load

When users select this option, a *list-request* event is primarily sent through the socket. Within the related event listener, the server will send back a *file-list* event to the application, containing the names of the files stored in the *saves* folder. These names are then shown as menu buttons. Clickable arrows, like the one described in Section 5.2.6, will allow to navigate through this list. Once a file is chosen, the web page will be simply reloaded, but with an additional query parameter that will hold, indeed, this file name. This parameter will be then accessed by the main component, during its preliminary operations. Instead of using the standard taxonomy file, the component will exploit this specific one to build the scene. Further, the *filename* box will hold the content of the query parameter in place of the usual *New project* label.
From then on, the application will behave according to the usual workflow.
Naturally, if an invalid parameter is manually inserted in the page address, an

error will be printed in the *log* box. In that case, the application will solely allow users to open this menu to properly load a valid file.

New

This option is equivalent to an *undo all* button. All non-saved changes made to the model so far will be discarded and the application will create a brand new session for users.
This has been implemented by simply reloading the page with no query parameters. In this way, the main component will build the scene by reading the default file used for the taxonomy.

Chapter 6

Results

6.1 Evaluation

After the development process, an exploratory study has been carried out to evaluate the designed platform. It has been based on experimental tests led at the Politecnico of Turin, involving 24 volunteers.
In particular, two conditions were compared: an immersive setup (IM) and a non-immersive one (NI), differing basically in perceived immersion and presence, relating to the hardware used to interact with the application. The first setup, indeed, required the use of an HMD along with tracked controllers, while the latter only involved mouse and keyboard on a traditional desktop environment.
In addition to assessing the quality of the designed application, the main goal of the evaluation was to investigate four hypotheses:

- *H1*: A desktop setup and an immersive one are equivalent in terms of performance when the active exploration of a large setting is not needed.
- *H2*: Concerning spatial learning tasks, immersive features can lead to better performances than those offered by a desktop environment.
- *H3*: When the interaction with a graphical user interface is required, a mouse-based approach still results in better performances than hand-tracked controllers.
- *H4*: An immersive setup provides users with a better experience than a desktop environment.

The participants were therefore divided into two separated groups and assigned to the setups since being exposed to the more appealing VR devices could have biased them to prefer the immersive condition over its traditional counterpart.
As in many other literary works, a task-based approach has been adopted to compare both the user performance and the user experience. For each task, performance was measured using two parameters: the completion time and the number of errors made. On the other hand, to examine how users perceived the experience and to uncover further subjective factors, a Likert-scale questionnaire has been used. Such a questionnaire aims to assess the overall usability in both scenarios, allowing also to compare critical factors. Moreover, the experienced simulator sickness, particularly meaningful to rate VR environments, has been recorded.
Three tasks, mainly focused on spatial learning and exploration, have been designed to test users and let them browse the functionalities of the application:

1. *Navigate to a specific term*: to introduce users to the basic mechanics of the designed application, they are firstly required to navigate up to a particular bubble. Rather than randomly wandering throughout the environment, they can use the vocal commands to know about the exact position of the term in the taxonomy hierarchy. The task is considered complete when the user steps into the requested bubble.

2. *Find the most frequent term*: by exploiting the visual design of the bubbles, users are then asked to actively explore the environment, comparing entities, to find the most frequent term within the taxonomy. Once found, it is easily detectable since it is characterized by a full green bar, representing its importance, right above itself. The task is considered complete when the user selects the term bubble.

3. *Create and edit a bubble*: to stimulate the users' commitment and to allow them to interact with the correcting interface, they are finally asked to add a new term to the visualization. They are free to create it inside the category they want, naming it as they please. Next, they have to give it a frequency, by editing it. In the end, users have to push out the created term from the current category and add it to another one. The interface will eventually recall to them that terms can not be pushed into other

terms. The task is considered complete once the user pushes the created term into the destination bubble.

During the tasks, furthermore, certain user behaviors have been noted down, to record their preferences and the different choices made. Specifically, the various methods used to navigate inside or outside a bubble were analyzed, together with the vocal commands exploited.
The following sections describe the experiment, from the apparatus used and the procedure followed, to the main obtained results.

6.2 Experimental settings

6.2.1 Apparatus

Both the conditions were tested using a medium-high performance PC running Windows 10.
For the first group, who experienced the IM setup, an *Oculus Rift* headset has been used. *Oculus Touch* controllers were also provided as input devices. The second group, who tried the NI approach, received no additional hardware.

6.2.2 Procedure

The experiment has been conducted in six different phases, involving one user at a time.
Initially, participants had to fill a pre-test questionnaire, to gather demographic data and assess their previous existing knowledge both of the related topics (e.g. data analytics) and the involved technologies.
Then, only for the IM setup, they were asked to rate several symptoms regarding simulator sickness. Naturally, this measure is not meaningful for a desktop environment.
Next, the evaluator trained the participants on how to navigate and interact with the 3D environment through the input devices. The several functionalities of the application were here fully explained. Users were left free to use the application until they felt ready to begin the test.
The fourth phase then implied that, iteratively, users executed a task and rated the perceived level of comfort through the questionnaire. The evaluator stayed with the participants to provide them assistance, taking also useful notes about the ongoing experience but did not interfere with the interaction

at all. A video record of each performance has been stored for further analysis. After having answered the questions regarding the last task, users who tried the IM setup had to rate again the symptoms mentioned during the second phase, to estimate the sickness caused by the VR application.
Subsequently, all participants were asked to complete the questionnaire, by filling out the remaining parts concerning the user experience.

6.2.3 Case Study

A semantic dataset on plant genetic manipulation, supplied by the DEI team, was used as the input model for the proposed evaluation.
The related taxonomy contains more than 200 terms. This moderately crowded scenario has been chosen to stress differences between the two setups when it comes to detecting specific bubbles.
Moreover, the term importance is distributed in such a way that it is not so trivial to find the most frequent term. Highly frequent terms, indeed, are scattered through categories together with the rarest ones. In this manner, the category in which the most important term resides, will not necessarily be the largest.

6.2.4 Questionnaire

The used questionnaire can be consulted in Appendix A.
It has been realized following other standard surveys, already used to rate user experience in similar contexts. Specifically, SUS[50], USE[51], VRUSE[52], SSQ[53], and Witmer Presence questionnaire [54] have been exploited.
The questionnaire has been also split into several sections grouping related questions. It has been further kept general enough to be filled by users experiencing one condition or the other. Besides an early part, aiming to gather general information on participants and their pre-existing knowledge, the other arranged sections are:

- Pre-test simulator sickness symptoms rating
- Task rating
- Post-test simulator sickness symptoms rating
- Overall usability

- Usability factors comparison

In the final section, through open questions, users were also required to point out three aspects that they appreciated and three features that they would have changed.

6.3 Participants

The 24 volunteers who joined the experimental tests were all university students, mainly recruited from biological sciences and engineering courses. They were 8 females and 16 males aged between 21 and 28 years old (see Figure 6.1). Participants have been randomly divided among the two groups, though gender proportions were preserved (4 females and 8 males).

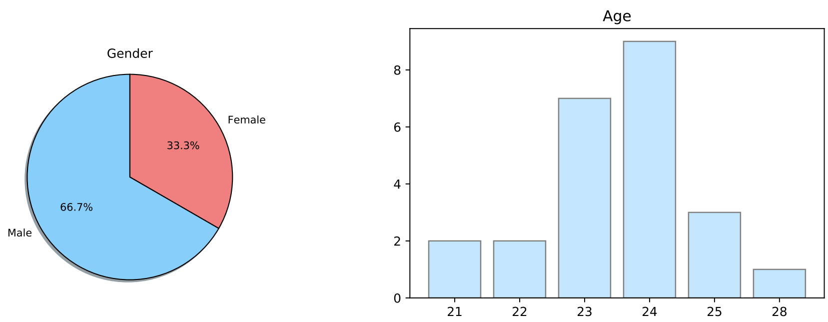


Figure 6.1: General information about the participants

As shown in Figure 6.2, most of the participants presented a good knowledge of desktop-based applications, claiming to use them often. Further, their mixed confidence with data analysis tools emerged. Conversely, they were, with very few exceptions, poorly familiar with the concept of Virtual Reality and their previous experience with HMDs and tracked controllers was basically nothing.

6.4 Results and findings

After the experiments, test data were collected, plotted, and further analyzed. The results of the study will be examined and discussed hereinafter, one section for each considered measure.

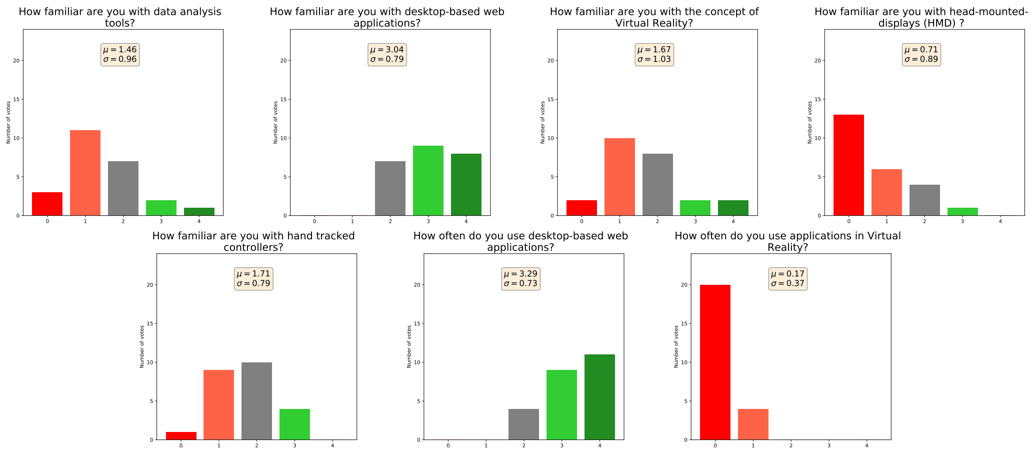


Figure 6.2: Pre-existing knowledge of the participants

The Student's (unpaired two-sample) T-test has been used to compare the mean values of the collected answers among the two groups. This has been done to check whether differences between the setups had statistical significance or not. Specifically, given the *p-value* obtained from the test, two questions were assumed to be *significantly different* if $p < 0.05$. Moreover, differences have to be considered *very significant* if $p < 0.01$ and *extremely significant* if $p < 0.001$.
Whenever no significant difference was detected, the results acquired from the two groups have been put into a single chart. Questions that received significantly different responses, on the contrary, have been marked with an asterisk (*) and the related results were put in two distinct charts. Specifically, the above chart always refers to the NI setup, while the one below relates to the IM setup.

6.4.1 Simulator sickness

As mentioned, the SSQ questionnaire has been used to assess and identify simulator sickness. It asks participants to rate 16 symptoms on a 0-3 value range. These symptoms can be traced back to three general factors: oculomotor (O), disorientation (D), and nausea (N). Each symptom value may or may not weight these categories. Hence, weighted values are summed up to obtain a score for each category, plus an overall total score (TS) [53].
The analysis of the symptoms in the IM setup before and after the experience (Figure 6.3), showed no significant difference and factor scores were found to

be relatively low. Only in a few cases slight fatigue, eye strain, and headache were recorded after the experience, maybe due to an effort made to read labels, menus, and instructions. None of the nausea-related symptoms, usually tied to the provided motion system, seemed to increase. An high standard deviation was reported, though, since only one participant suffered slight symptoms both before and after the tests.
Anyway, none of them experienced a simulator sickness such as to compromise their performance during the tasks.

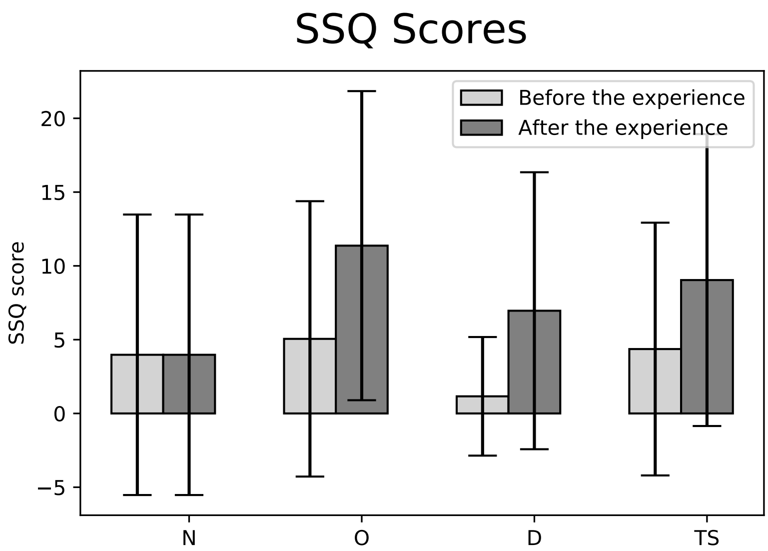


Figure 6.3: SSQ mean scores and standard deviations for the experiment

6.4.2 SUS score

The SUS questionnaire is widely used to evaluate the general usability of a system and the satisfaction felt by users during its use. It consists of 10 questions rated in a 0-4 value range. These values are then converted and summed up to obtain a 0-100 total score for each participant [50].
As can be observed in Figure 6.4, the two setups reported almost no difference. In particular, the IM setup obtained an average score of 86.4, while the NI setup scored an average of 82.8. This shows both a slight superiority of the IM setup and the general good reception of the application.

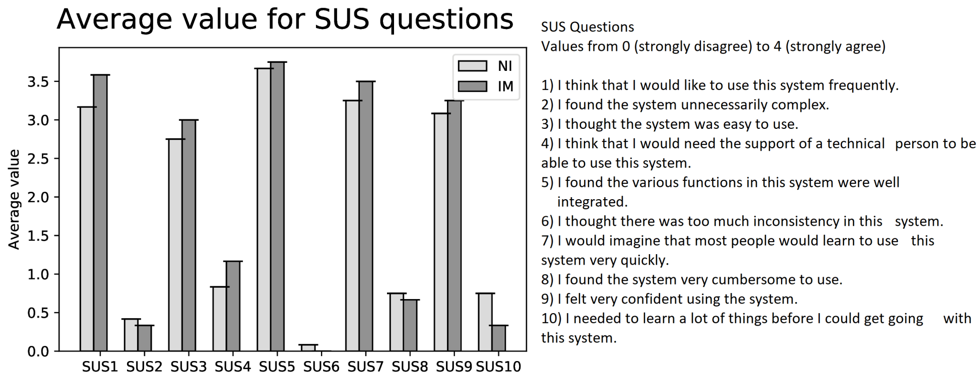


Figure 6.4: Questions and related average responses for the SUS questionnaire

6.4.3 User performance

The user performance was analyzed by measuring both completion times and errors made. During the experience, furthermore, the various methods and features exploited by participants to perform certain actions were noted down, to detect their preferences.

Completion time

Figure 6.5 shows the differences in time taken to complete the tasks between the two groups.
Concerning the first task, the time analysis resulted in $p = 0.36$, showing that, when the final destination was known, both conditions required more or less the same time to reach it. Also, the vocal research of a term required the same time and worked as expected.
For the second task, instead, NI ($M = 124$, $SD = 24.17$) and IM ($M = 46.17$, $SD = 30.32$) values with $p < 0.001$, show that the latter condition substantially advantaged users during the exploration task. Most of the users in the NI setup, indeed, had to unsuccessfully explore several bubbles before finding the most frequent one, while the other group uniformly found the target after a few attempts, by exploiting the size of the spheres. Maybe this was due to a better-perceived sense of depth and scale of the scene.
Regarding the third task, NI ($M = 132.75$, $SD = 22.09$) outperformed IM ($M = 242$, $SD = 50.25$) with $p < 0.001$. This is attributable to a significantly faster menu navigation provided by mouse and keyboard compared to the controllers. It has been mainly caused by higher confidence from users with the former

devices, while they acted more carefully and slowly with the latter ones. Such a result is totally consistent with H3.

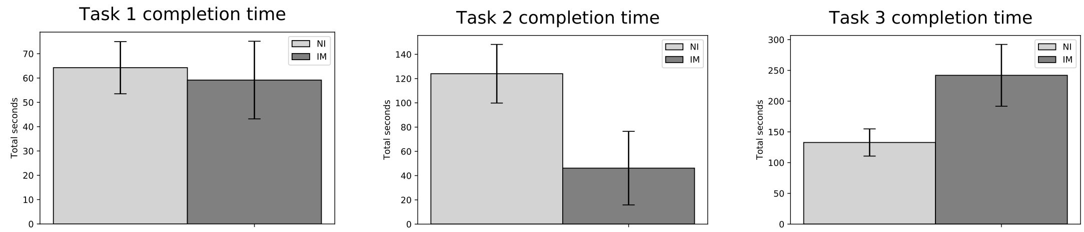


Figure 6.5: Mean values and standard deviations of the total time spent to complete each task.

Errors made

During each task, any unwanted action from users (e.g. involuntarily opening a menu or exiting/entering from a bubble), along with the ones not strictly required (e.g. navigating to the wrong term, exploring the smallest bubbles, picking the wrong menu option) were considered as errors.
Figure 6.6 shows the difference in made mistakes among the two setups. The compared conditions appeared equivalent in terms of errors and no statistically significant difference was found. The obtained p-values, indeed, are respectively $p1 = 0.45$, $p2 = 0.29$, $p3 = 0.54$.
Within the first task, users in both setups stepped in a wrong bubble or opened some menus unintentionally, still trying to become acquainted with the input devices. However, as previously assumed by H1, the overall user performances concerning this basic navigation task do not appear to be influenced by the condition in which the application is used.
Concerning the second task, some of the participants lingered in the exploration of certain bubbles which were sensibly too small to contain the most frequent term, but this happened more often to the NI group. This result, together with the analysis of completion time here above, seems to confirm H2.
Lastly, as suggested by the number of errors reported in the last task, even if the NI condition enabled significantly faster interactions, laser-shooting controllers proved themselves to be slightly more precise in selecting options.

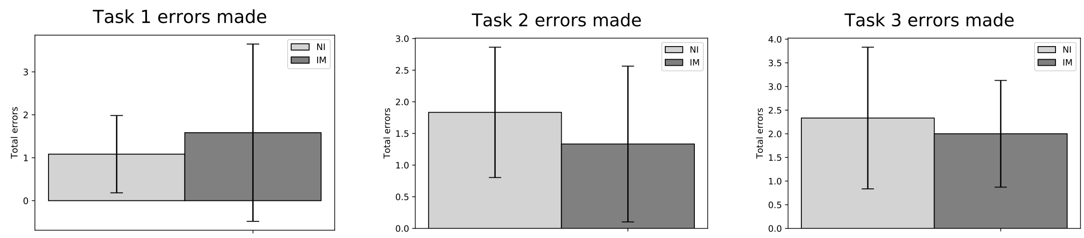


Figure 6.6: Mean values and standard deviations of the errors made by the users in each task.

Feature preferences

As has been evidenced by the analysis of the user sessions, the different methods provided to navigate the environment were exploited very differently by the two groups. Users of NI, though could still attract bubbles and click on the dome to rapidly travel among levels, kept instinctively preferring to walk around the environment with WASD keys. Further, in this condition, the interactive breadcrumb was mainly neglected and used only when strictly necessary (i.e. to quickly travel among non-adjacent layers). Moreover, they used the vocal commands much more often than in IM, especially to attract bubbles and travel back, explicitly reporting to have sensibly appreciated this feature. Conversely, the IM users moved much more rapidly, attracting bubbles and using the teleportation. They appeared to appreciate even more the breadcrumb menu, often used also to travel between adjacent levels. In this setup, the vocal commands were more ignored in favor of controller-based interactions.

6.4.4 User experience

The quality of the user experience has been quantified through the questionnaire responses. They are all expressed on a 5-grade Likert scale, varying from *strongly disagree* to *strongly agree.* In this way, it was possible to estimate the perceived comfort during each task and the application usability, both in general and regarding specific factors. Moreover, overall questions were used to allow users to rate every examined feature from *very unsatisfactory* to *very satisfactory.*

Task rating

As can be seen in Figure 6.7, the first task has been perceived similarly in the two conditions. It was generally considered to be trivial to perform and no one felt frustrated during its execution. Consequently, the resulting overall satisfaction appeared to be quite limited.

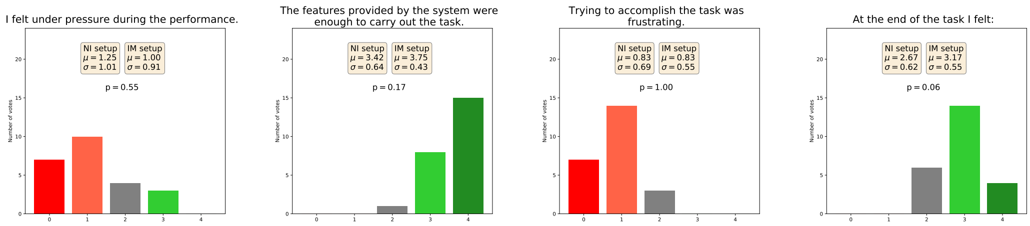


Figure 6.7: User responses regarding their experienced level of comfort during Task 1

Regarding the second task, instead, users' opinions were significantly different among the two groups (see Figure 6.8). Some of them perceived the necessity of finding such an hidden information as a preoccupation, worried that they could not succeed. Some others rather saw the task as a stimulating challenge and reported no perceived pressure. Furthermore, due to the larger time spent, some of the NI users casted doubts on the appropriateness of the features provided to carry out the task. They also reported a mixed level of frustration and a contained overall satisfaction. The IM group, conversely, rated the system to be adequate for the task. Almost unanimously, they did not experience any frustration and felt very satisfied. This is probably due to their higher capacity of perceiving the relative size of the elements, allowing to efficiently carry out the task.

The third task, like the previous one, divided the audience (see Figure 6.9). Most of the NI group felt no pressure and their final satisfaction was not so high. Probably they found the task too simple to perform. On the other hand, menu navigation through laser controllers appeared to intimidate those in the IM condition. Maybe as a consequence, they felt much more satisfied when eventually succeeded. The recorded level of frustration appears to be very diversified among users of the two setups. Both groups, though, agreed that the provided control interface was enough to carry out the task.

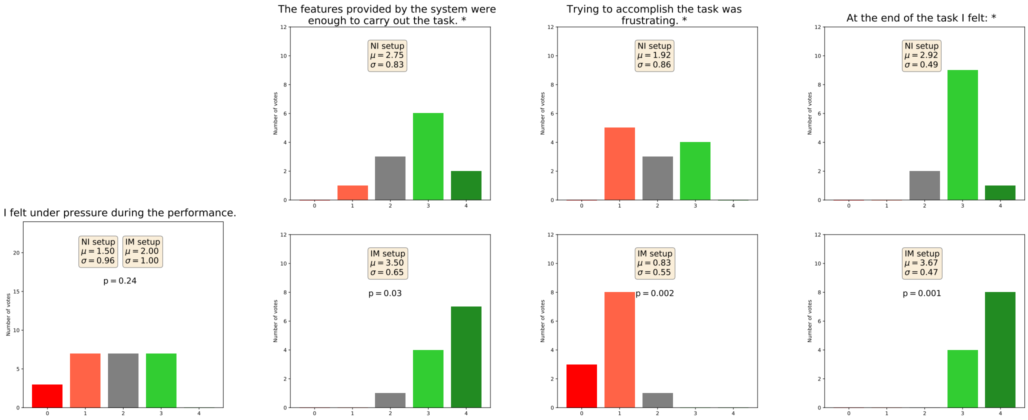


Figure 6.8: User responses regarding their experienced level of comfort during Task 2

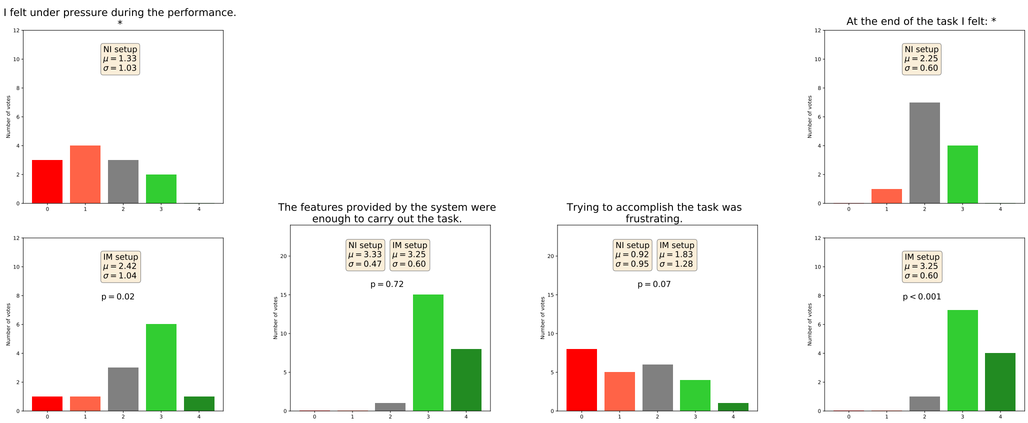


Figure 6.9: User responses regarding their experienced level of comfort during Task 3

Overall usability

As shown in Figure 6.10, the answers concerning overall usability highlights a good reception in terms of perceived ease of use, experienced errors, and general satisfaction for both conditions. Participants mainly agreed or strongly agreed to all those questions asking to rate the appropriateness, utility and comprehension of the user interface. Similarly, they disagreed or strongly disagreed to all those statements about its weaknesses.

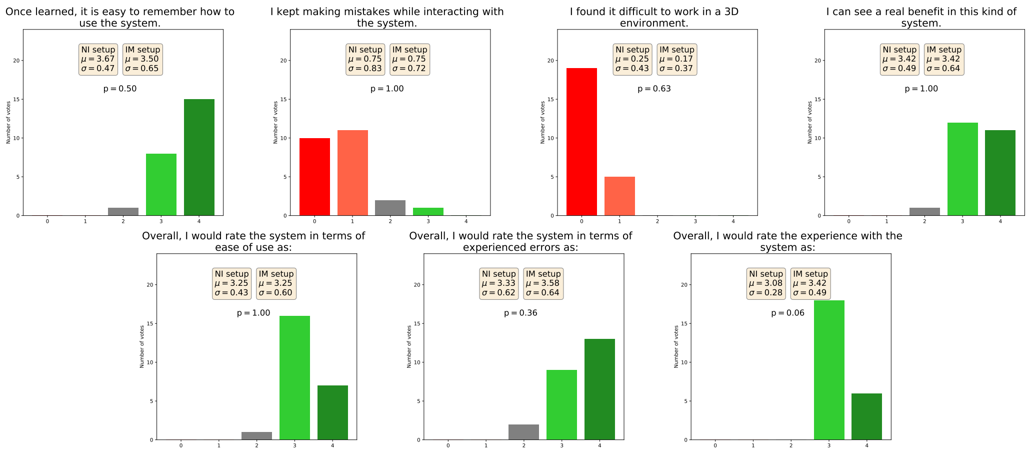


Figure 6.10: Non-significantly different responses regarding overall user experience

In Figure 6.11, instead, are depicted the questions that received significantly different responses within the two groups. Remarkably, the users appeared to be sensibly more enthusiastic about the IM setup, even though they diffusely disagreed on the possibility of using it for long periods. Indeed, they claimed that prolonged exposure to the application in VR could disorientate, cause them a headache and strain their eyes.

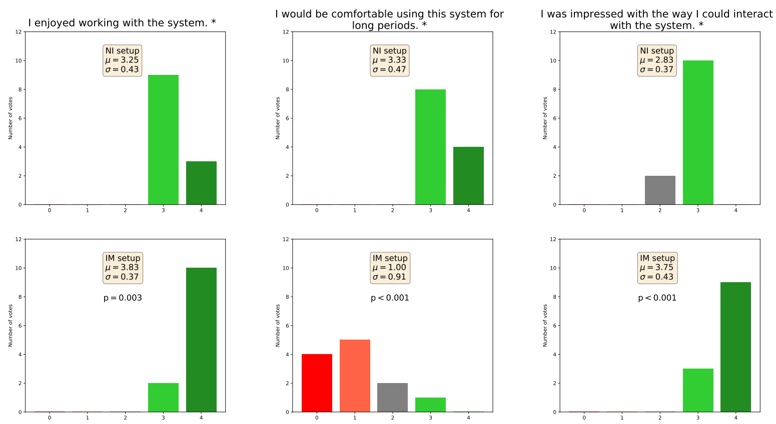


Figure 6.11: Significantly different responses regarding overall user experience

Functionality

The answers about the functionality level of the application (see Figure 6.12) shows a general strong appreciation from both groups. The various interfaces were found to be intuitive and easy to use. Notably, a significant difference was reported on the question regarding the features used throughout the experience. The NI group has shown itself to be widely divided on the topic, while the IM users agreed that they neglected some of them. As mentioned before, indeed, some elements like the vocal commands and the breadcrumb menu were used and appreciated differently in the examined conditions.

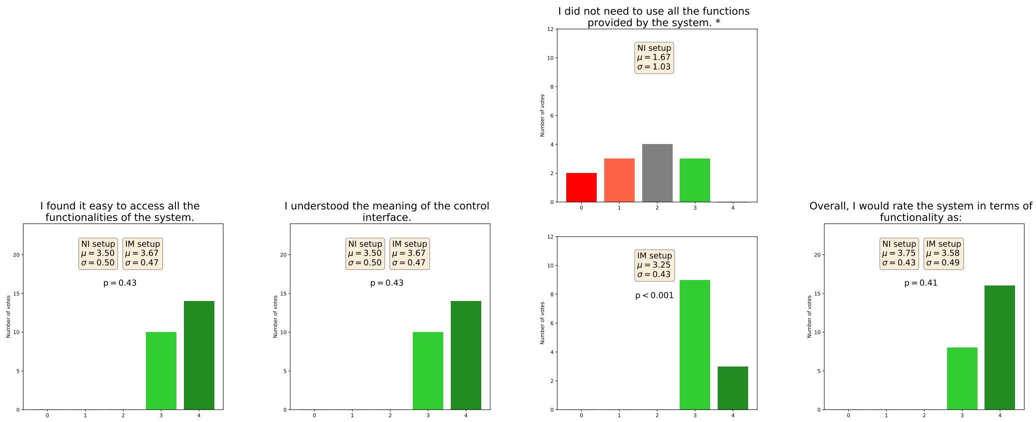


Figure 6.12: User responses regarding the functionality level

User input

Figure 6.13 depicts the responses on the input devices. In both conditions, they were generally perceived as easy to use and adequately integrated into the application. This result was expected for mouse and keyboard but was not so evident for the tracked controllers. This allowed users to concentrate more on the required tasks. Also, the environment navigation system provided, in both cases, was found to be generally appropriate. Some flaws have been encountered though and this led most of the users to rate the input as good but not optimal. Specifically, users in the IM group complained about the high sensitivity of the controllers, especially the thumbsticks. More than once, indeed, they unintentionally opened the hand-attached menus. Nevertheless, most of those users appreciated the natural way to interact with bubbles through controllers. On the other hand, opinions about the click-based mechanism provided in the NI setup have not been uniform. Some participants immediately familiarized themselves with such a feature, while some others initially struggled to use it. However, most of the users soon became acquainted with it and did not experience any particular issue.

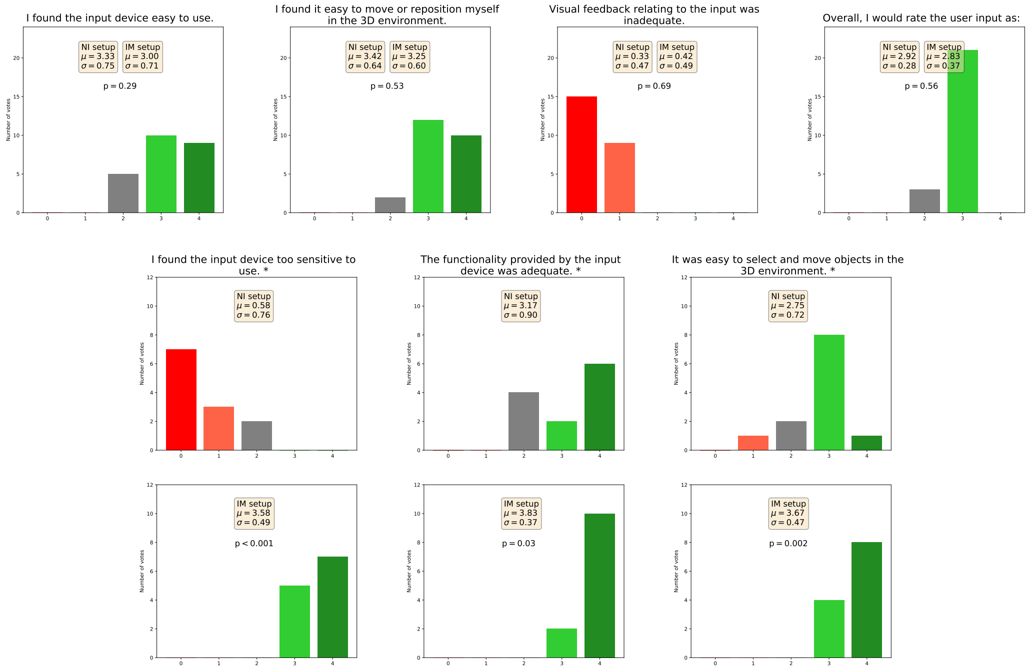


Figure 6.13: User responses regarding the input devices

System output

As can be seen in Figure 6.14, the displayed output was highly regarded by both groups, with almost no differences. Users did not experience any delay in the image, nor any significant distortion. Generally, they all agreed that the quality of the output did not affect their performance. Furthermore, they found how information was presented very meaningful and not even a bit complicated. As it had already been highlighted from the SSQ outcomes, both groups did not feel nauseous at all. The only significant difference found regards the field of view. While IM users have been very satisfied with it, the NI group's opinion was less warm and, rather, some complained. They later stated that they frequently had to drag the cursor around to move the view and that this was slightly annoying, especially during the exploration task. Despite this, they overall rated the system output as largely satisfactory.

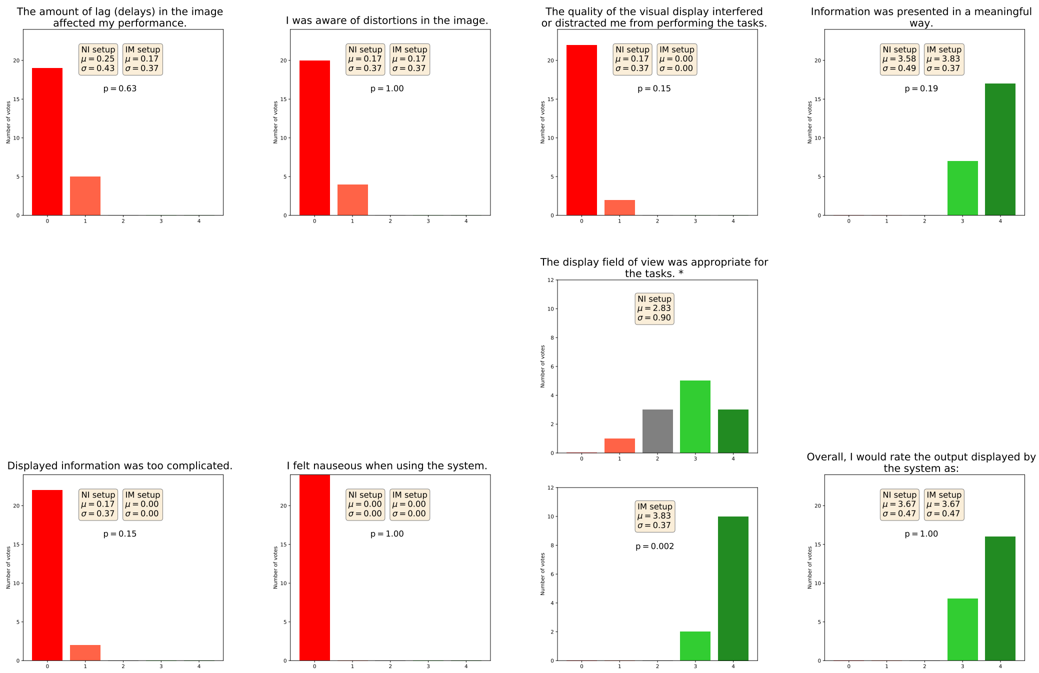


Figure 6.14: User responses regarding the system output

User guidance and help

Regarding the provided guidance and help, users in both groups agreed that the various menu options and the content of text boxes were really easy to understand. On the other hand, they showed very different opinions on the way some functionalities were accessed. Specifically, during the first phases of the experience, several IM users struggled to remember the button positions on tracked controllers and the functionalities related to them. They also explicitly suggest, as a future improvement, to add a visual panel to remind them of this information. Anyway, they became proficient after a bit of training. Conversely, NI users did not have similar problems. Moreover, most of the users noted that the provided instructions should have been displayed on a detached window, rather than on the log box, which may be limiting for long messages. Due to this, they generally rated the guidance system as satisfactory although improvable.
These outcomes are depicted here below in Figure 6.15.

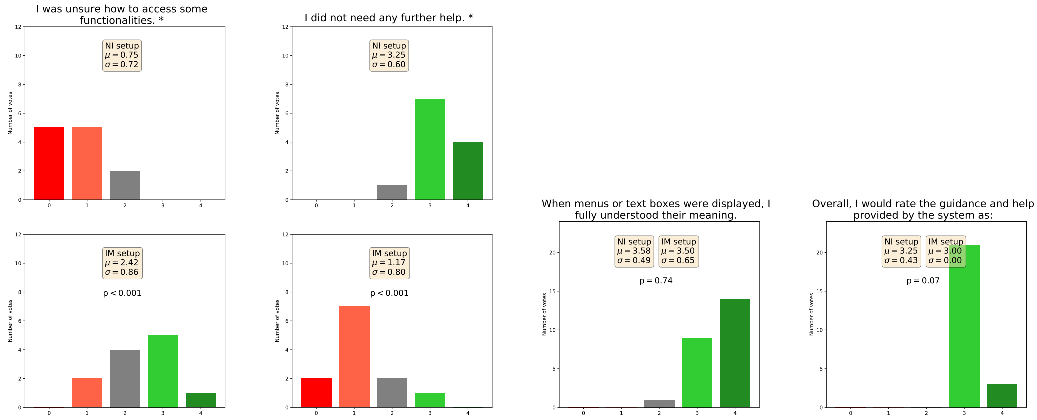


Figure 6.15: User responses regarding the provided guidance and help

Consistency

As is depicted in Figure 6.16, most of the users agreed on the good consistency level of the application. They did not reported any particular unexpected behavior, both in the scene and in the operations triggered by their input.

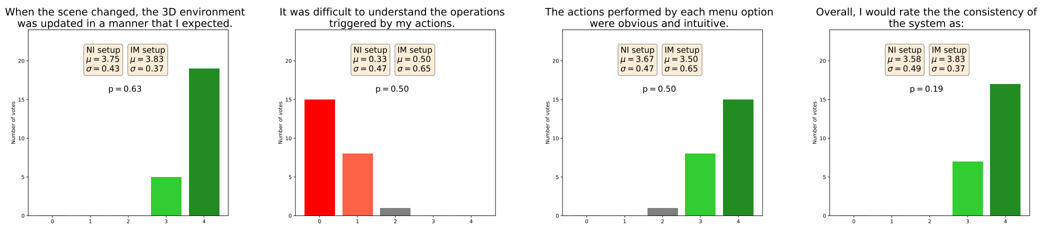


Figure 6.16: User responses regarding the consistency of the system

Flexibility

Also the flexibility of the system was highly rated (see Figure 6.17). Both groups appreciated that different methods were available to perform the required actions. No one complained about not being able to achieve something specific. Moreover, users were very satisfied with examining elements at different distances and viewpoints.

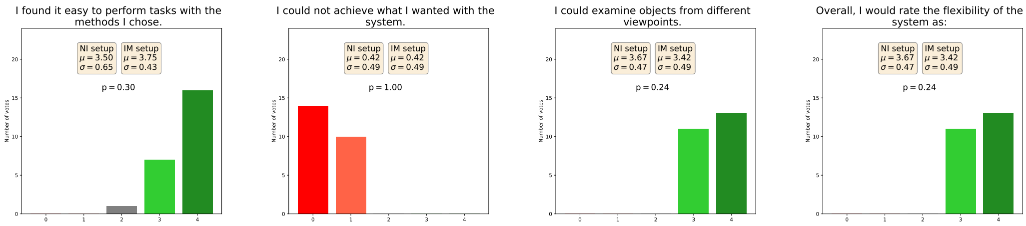


Figure 6.17: User responses regarding the flexibility of the system

Error correction and handling

Similarly to flexibility and consistency, also the facilities provided to handle and correct errors were appreciated in both setups with no significant differences. This can be observed in the responses depicted in Figure 6.18. Generally, users noticed when they made an error. They also found it easy to undo their mistakes and never remained stuck to the point that they could not go ahead.

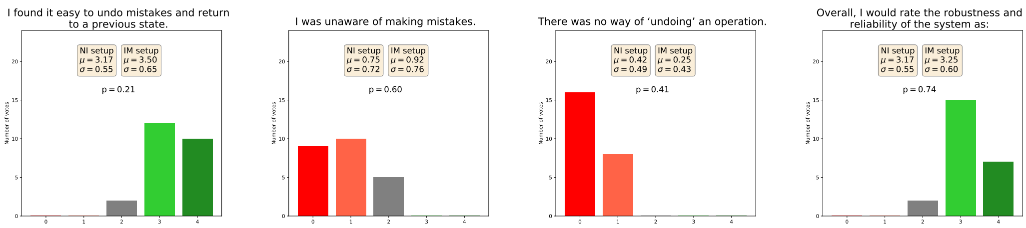


Figure 6.18: User responses regarding the possibilities of error correction and handling

Spatial involvement

As shown in Figure 6.19, the main differences among the two setups were detected through the last section of the questionnaire, about the relationship between users and the surrounding environment. As expected, the IM setup provided them with a very good sense of presence, to the extent that they agreed to have lost track of the time. They also confirmed to have no idea of what was going on around them, due to the worn HMD. It was found that the majority of them felt the immersion necessary to carry out the tasks. Moreover, they agreed that an higher sense of immersion could have further sped up their performance.

Naturally, NI participants mostly disagreed when asked if they felt to be within the environment or if they were so absorbed by the application. Furthermore, they interestingly believed that such an immersion was not even necessary and, as some of them stated, could have been a further source of distraction. The desktop users also lamented a lack of depth in the image to have influenced their performance, essentially referring to the second task. Participants found the WASD-based movement very natural to explore the environment. The teleportation system was rated to be quite natural too, but some IM users stated that they found a bit cumbersome pointing the lasers behind their back to move backward.

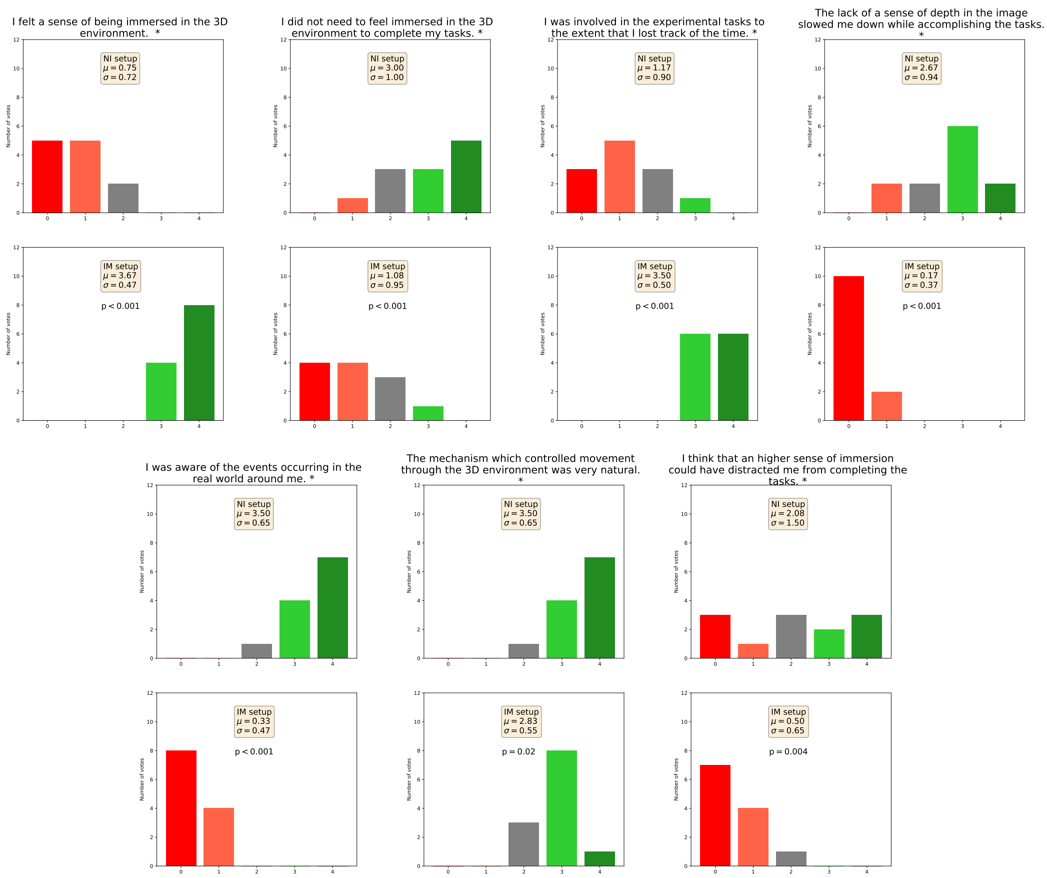


Figure 6.19: Significantly different responses regarding spatial involvement

Figure 6.20, instead, shows the few questions for which the responses resulted in the same trend. In both conditions, participants learned how to proficiently move and interact. It has been also reported that they constantly knew the path followed, thanks to the related text box, and never felt lost. Hence, the relationship with the environment was generally rated as satisfactory.

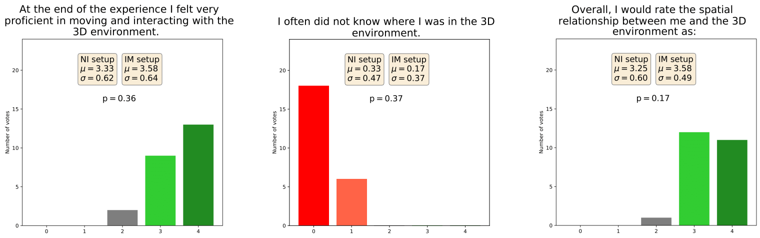


Figure 6.20: Non-significantly different responses regarding spatial involvement

In conclusion, such an analysis of the user experience on both setups, together with the earlier SUS score comparison, has revealed a general superiority of the immersive condition over the traditional desktop environment, confirming the last research hypothesis (H4). Nevertheless, this latter approach has been mostly appreciated and exhibited specific advantages too.

Chapter 7

Conclusions and perspectives

7.1 Conclusions

The present thesis work led to the realization of a tool to support the analysis of large semantic datasets built from collections of documents. Such an application has been developed taking into account accessibility and portability, and was therefore distributed on the Web. According to the Immersive Analytics research agenda, modern VR technologies have been chosen as the main interface to increase user engagement and, hence, to stimulate the study of several topics. A desktop-based version has been also designed to exploit the tool even when a VR headset was not available. A comparative evaluation of these two conditions was then conducted.
The analysis of the experimental results obtained after the evaluation phase proved the overall quality of the application in both setups. The participants appeared to be generally satisfied and stated that this kind of system had aroused their interest. In particular, through some final open questions, it has been possible to detect which aspects they valued the most and which were the ones that could be improved.
Many of them appreciated the insight-stimulating visual design of the presented scene. They claimed it was simple but very intuitive, universal, and effective at the same time. The menu interface that allows to modify the underlying model was also mentioned as a very interesting feature. The possibility to move elements around and to explore different levels was also largely

welcomed. Moreover, participants have shown themselves particularly enthusiastic about the vocal commands and the teleport-based navigation system. Among the limitations of the designed tool, many users of the desktop version mentioned the counterintuitive separation between mouse and cursor (i.e. the necessity to drag and drop), and the lack of a way to run, besides walking around. Regarding the immersive setup, instead, most of the complaints were on the difficulty of remembering the functionality of each button and the high sensitivity of the controller thumbsticks. They suggested adding a floating instruction panel depicting the model of the controller and, contextually, the actions performed by each button. Users in both groups also stated that it was not so trivial to understand what was the exact distance at which the animation to enter a bubble was triggered. It was suggested, for instance, to provide users with sound effects as feedback for each specific interaction.

7.2 Perspectives

One of the main future improvements may consist of adding an interface that allows users to directly load their custom corpus of documents to generate a brand new taxonomy from scratch. In such a way, the application would go beyond its original purpose of supplying mere visualization, becoming a stand-alone analysis tool. Since the current system is already distributed, a web application like the one realized by Díaz et al. [46] could be easily integrated.

Moreover, several users suggested to enhance some aspects of the graphic design. In particular, they recommended a more striking style for text labels and proposed to employ more vivid cartoon-style colors. It has been also pointed out that some other kind of media could be added within each term to supply additional details, such as audio/video sources and animated 3D models. Other interesting suggestions have been about providing a minimap depicting a summary of the environment and giving also the possibility to switch between different camera views (e.g. an inside-the-data and a bird's eye perspective), allowing to pass from an highly specific level of detail to a general overview in a very flexible way. An user also suggested exploiting vocal commands to enable more complex queries, through, for instance, a mechanism based on questions and answers. This should automate some of the tasks that must be currently performed by humans, like finding the most notable words along with outliers.

Appendix A

Evaluation questionnaire

Gender:
Age:
Occupation:
Pre-existing knowledge
Values from 0 (none/never) to 4 (a lot/very often):

- How familiar are you with data analysis tools?
- How familiar are you with desktop-based web applications?
- How familiar are you with the concept of Virtual Reality?
- How familiar are you with head-mounted-displays (HMD)?
- How familiar are you with hand tracked controllers?
- How often do you use desktop-based web applications?
- How often do you use applications in Virtual Reality?

Simulator sickness (PRE-TEST)
Rate each symptom before performing the test.
(Value range 0-3: none, slight, moderate, severe)

- General discomfort
- Fatigue
- Headache
- Eye strain
- Difficulty focusing
- Increased salivation
- Sweating

- Nausea
- Difficulty concentrating
- Fullness of head
- Blurred vision
- Dizzy (eyes open)
- Dizzy (eyes closed)
- Vertigo
- Stomach awareness
- Burping

Task 1 rating
(Value range 0-4: strongly disagree, disagree, neutral, agree, strongly agree)

- I felt under pressure during the performance.
- The features provided by the system were enough to carry out the task.
- Trying to accomplish the task was frustrating.
- At the end of the task I felt:
  very unsatisfied, unsatisfied, neutral, satisfied, very satisfied.

Task 2 rating
(Value range 0-4: strongly disagree, disagree, neutral, agree, strongly agree)

- I felt under pressure during the performance.
- The features provided by the system were enough to carry out the task.
- Trying to accomplish the task was frustrating.
- At the end of the task I felt:
  very unsatisfied, unsatisfied, neutral, satisfied, very satisfied.

Task 3 rating
(Value range 0-4: strongly disagree, disagree, neutral, agree, strongly agree)

- I felt under pressure during the performance.
- The features provided by the system were enough to carry out the task.
- Trying to accomplish the task was frustrating.
- At the end of the task I felt:
  very unsatisfied, unsatisfied, neutral, satisfied, very satisfied.

Simulator sickness (POST-TEST)
Rate each symptom after having performed the test.
(Value range 0-3: none, slight, moderate, severe)

- General discomfort

- Fatigue
- Headache
- Eye strain
- Difficulty focusing
- Increased salivation
- Sweating
- Nausea
- Difficulty concentrating
- Fullness of head
- Blurred vision
- Dizzy (eyes open)
- Dizzy (eyes closed)
- Vertigo
- Stomach awareness
- Burping

OVERALL USABILITY
Ease of use
(Value range 0-4: strongly disagree, disagree, neutral, agree, strongly agree)

- I thought the system was easy to use.
- I think that I would need the support of a technical person to be able to use this system.
- I would imagine that most people would learn to use this system very quickly.
- I found the system very cumbersome to use.
- I needed to learn a lot of things before I could get going with this system.
- Once learned, it is easy to remember how to use the system.
- Overall, I would rate the system in terms of ease of use as:
  very unsatisfactory, unsatisfactory, neutral, satisfactory or very satisfactory.

Experienced errors
(Value range 0-4: strongly disagree, disagree, neutral, agree, strongly agree)

- I thought there was too much inconsistency in this system.
- I felt very confident using the system.
- I could recover from mistakes quickly and easily.

- I kept making mistakes while interacting with the system.
- I was confused by the operation of the system.
- Overall, I would rate the system in terms of experienced errors as:
  very unsatisfactory, unsatisfactory, neutral, satisfactory or very satisfactory.

Satisfaction
(Value range 0-4: strongly disagree, disagree, neutral, agree, strongly agree)

- I enjoyed working with the system.
- I found it difficult to work in a 3D environment.
- I think that I would like to use this system frequently.
- I would be comfortable using this system for long periods.
- I found the system unnecessarily complex.
- I found the various functions in this system were well integrated.
- I was impressed with the way I could interact with the system.
- I can see a real benefit in this kind of system.
- I think that using the system could improve my knowledge on a topic.
- Overall, I would rate the experience with the system as:
  very unsatisfactory, unsatisfactory, neutral, satisfactory or very satisfactory.

SPECIFIC USABILITY FACTORS
Functionality
(Value range 0-4: strongly disagree, disagree, neutral, agree, strongly agree)

- I found it easy to access all the functionalities of the system.
- The level of functionality was appropriate for accomplishing the tasks.
- I understood the meaning of the control interface.
- I did not need to use all the functions provided by the system.
- Overall, I would rate the system in terms of functionality as:
  very unsatisfactory, unsatisfactory, neutral, satisfactory or very satisfactory.

User input
(Value range 0-4: strongly disagree, disagree, neutral, agree, strongly agree)

- I found the input device easy to use.
- I found the input device too sensitive to use.
- I could concentrate more on the required tasks than on the mechanisms to perform them.
- The functionality provided by the input device was adequate.

- It was easy to select and move objects in the 3D environment.
- I found it easy to move or reposition myself in the 3D environment.
- Visual feedback relating to the input was inadequate.
- Overall, I would rate the user input as:
  very unsatisfactory, unsatisfactory, neutral, satisfactory or very satisfactory.

System output
(Value range 0-4: strongly disagree, disagree, neutral, agree, strongly agree)

- The display field of view was appropriate for the tasks.
- The amount of lag (delays) in the image affected my performance.
- I was aware of distortions in the image.
- The quality of the visual display interfered or distracted me from performing the tasks.
- Information was presented in a meaningful way.
- Displayed information was too complicated.
- I felt nauseous when using the system.
- Overall, I would rate the output displayed by the system as:
  very unsatisfactory, unsatisfactory, neutral, satisfactory or very satisfactory.

User guidance and help
(Value range 0-4: strongly disagree, disagree, neutral, agree, strongly agree)

- I was unsure how to access some functionalities.
- The provided instructions were very informative.
- I did not need any further help.
- When menus or text boxes were displayed, I fully understood their meaning.
- Overall, I would rate the guidance and help provided by the system as:
  very unsatisfactory, unsatisfactory, neutral, satisfactory or very satisfactory.

Consistency
(Value range 0-4: strongly disagree, disagree, neutral, agree, strongly agree)

- When the scene changed, the 3D environment was updated in a manner that I expected.
- It was difficult to understand the operations triggered by my actions.
- The use of icons, menus and buttons was inconsistent.
- The actions performed by each menu option were obvious and intuitive.
- The restrictions imposed by the system were consistent with its functioning.

- Overall, I would rate the the consistency of the system as:
  very unsatisfactory, unsatisfactory, neutral, satisfactory or very satisfactory.

Flexibility
(Value range 0-4: strongly disagree, disagree, neutral, agree, strongly agree)

- I found it easy to perform tasks with the methods I chose.
- I could not achieve what I wanted with the system.
- The user interface interfered with the way I wanted to interact with the system.
- I could examine objects very closely.
- I could examine objects from different viewpoints.
- Overall, I would rate the flexibility of the system as:
  very unsatisfactory, unsatisfactory, neutral, satisfactory or very satisfactory.

Error correction and handling
(Value range 0-4: strongly disagree, disagree, neutral, agree, strongly agree)

- I found it easy to undo mistakes and return to a previous state.
- I was unaware of making mistakes.
- The system provided protection against trivial errors.
- There was no way of ‘undoing’ an operation.
- Overall, I would rate the robustness and reliability of the system as:
  very unsatisfactory, unsatisfactory, neutral, satisfactory or very satisfactory.

Spatial involvement
(Value range 0-4: strongly disagree, disagree, neutral, agree, strongly agree)

- I felt a sense of being immersed in the 3D environment.
- I did not need to feel immersed in the 3D environment to complete my tasks.
- I was involved in the experimental tasks to the extent that I lost track of the time.
- The lack of a sense of depth in the image slowed me down while accomplishing the tasks.
- At the end of the experience I felt very proficient in moving and interacting with the 3D environment.
- I was aware of the events occurring in the real world around me.
- The mechanism which controlled movement through the 3D environment was very natural.
- I think that an higher sense of immersion could have distracted me from completing the tasks.

- I often did not know where I was in the 3D environment.
- Overall, I would rate the spatial relationship between me and the 3D environment as:
  very unsatisfactory, unsatisfactory, neutral, satisfactory or very satisfactory.

Open questions

- Describe 3 features that you liked.
- Describe 3 features that you would like to improve.

Bibliography

[1] P. Milgram, H. Takemura, A. Utsumi, F. Kishino, ”Augmented reality: A class of displays on the realityvirtuality continuum”, Photonics for industrial applications, pp. 282-292, 1995.
[2] Fagan, Kaylee. ”Here’s what happens to your body when you’ve been in virtual reality for too long”. Business Insider. Retrieved 5 September 2018.
[3] Mukamal, Reena (2017-02-28). ”Are Virtual Reality Headsets Safe for Eyes?”. American Academy of Ophthalmology. Retrieved 11 September 2018.
[4] Caddy, Becca (2016-10-19). ”Vomit Reality: Why VR makes some of us feel sick and how to make it stop”. Wareable.com. Retrieved 11 September 2018.
[5] Slater, M., Lotto, B., Arnold, M.M., Sanchez-Vives, M.V.: ”How we experience immersive virtual environments: the concept of presence and its measurement.” Anuario de psicología/The UB Journal of psychology 40(2), 193–210 (2009)
[6] C. Cruz-Neira, D. Sandin, T. DeFanti, R. Kenyon and J. Hart, ”The CAVE: audio visual experience automatic virtual environment,” in Commun. ACM 35(6), 1992, p. 64–72.
[7] Havig, P., Mcintire, J., Geiselman, E., 2011. Virtual reality in a cave: limitations and the need for HMDs? Proceedings of SPIE, 8041(1), pp.804107–804106.
[8] I. Sutherland, ”A head-mounted three dimensional display,” in Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I, ACM, 1968, p. 757–764.
[9] I. Sutherland, ”Sketchpad a man-machine graphical communication system,” in Transactions of the Society for Computer Simulation 2(5), R-3, 1964.
[10] S. Card, G. Robertson and J. Mackinlay, ”The information visualizer, an

information workspace," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 1991, pp. 181-186.
[11] J. Bertin, in Semiology of graphics: diagrams, networks, maps, University of Wisconsin press, 1983.
[12] J. Tukey, "Exploratory data analysis", Reading, Mass, 1977.
[13] E. Tufte, "Envisioning information", Graphics press, 1990.
[14] S. Card, J. Mackinlay and B. Shneiderman, in Readings in information visualization: using vision to think, Morgan Kaufmann, 1999.
[15] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. Stolper, M. Sedlmair, J. Chen, T. Möller and J. Stasko, "vispubdata.org: A metadata collection about IEEE visualization (VIS) publications," in IEEE Transactions on Visualization and Computer Graphics 23(9), 2017, p. 2199–2206.
[16] A. Cockburn and B. McKenzie, "An evaluation of cone trees," in People and Computers XIV – Usability or Else!, London, Springer, 2000, p. 425–436.
[17] A. Cockburn and B. McKenzie, "Evaluating spatial memory in two and three dimensions," in Int. J. Hum. Comput. Stud. 61(3), 2004, p. 359–373.
[18] J. Thomas and K. Cook, "Illuminating the path: the research and development agenda for visual analytics," in National Visualization and Analytics Ctr, 2005.
[19] Chandler, T., Cordeil, M., Czauderna, T., Dwyer, T., Glowacki, J., Goncu, C., . . . Wilson, E. (2015). Immersive Analytics. 2015 Big Data Visual Analytics (BDVA), 1-8.
[20] T. Itoh, K. Marriott, F. Schreiber and U. Wössner, "Immersive analytics: a new multidisciplinary initiative to explore future interaction technologies for data analytics," in Shonan Reports, 2016.
[21] T. Dwyer, N. Riche, K. Klein, W. Stuerzlinger and B. Thomas, "Immersive analytics (Dagstuhl seminar 16231)," in Dagstuhl Reports 6(6), 2016, pp. 1-9.
[22] Dwyer, T., Marriott, K., Isenberg, T., Klein, K., Riche, N., Schreiber, F., . . . Tobias, I. (2018). Immersive Analytics: An Introduction. In Immersive Analytics (pp. 1-23). Springer.
[23] Hashem, I., Yaqoob, I., Anuar, N., Mokhtar, S., Gani, A., Ullah Khan, S. (2015). The rise of "big data" on cloud computing: Review and open research issues. Information Systems, 47(C), 98-115.
[24] Lidong Wang, Guanghui Wang, and Cheryl Ann Alexander. 2015. Big

Data and Visualization: Methods, Challenges and Technology Progress. Digit. Technol. 1, 1 (2015), 33–38.

[25] B. Shneiderman, ”The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations,” in The Craft of Information Visualization, 2003, p. 364–371.

[26] Usher, W., Klacansky, P., Federer, F., Bremer, P., Knoll, A., Yarch, J., . . . Pascucci, V. (2018). A Virtual Reality Visualization Tool for Neuron Tracing. IEEE Transactions on Visualization and Computer Graphics, 24(1), 994-1003.

[27] Akira Kageyama, Yuichi Tamura, and Tetsuya Sato. 2000. Visualization of Vector Field by Virtual Reality stereo glasses wand. Prog. Theor. Phys. Suppl. 138 (2000), 665–673. DOI: https://doi.org/10.1143/PTPS.138.665 .

[28] Da Costa, L., Nedel, L. (2017). An Immersive Visualization Study on Molecules Manipulation. 2017 19th Symposium on Virtual and Augmented Reality (SVR), 2017, 169-177.

[29] Donalek, C., Djorgovski, S., Cioc, A., Wang, A., Zhang, J., Lawler, E., . . . Longo, G. (2014). Immersive and collaborative data visualization using virtual reality platforms. 2014 IEEE International Conference on Big Data (Big Data), 609-614.

[30] Wagner Filho, J., Rey, M., Freitas, C., Nedel, L. (2018). Immersive Visualization of Abstract Information: An Evaluation on Dimensionally-Reduced Data Scatterplots. 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 483-490.

[31] Heldal, I., Spante, M., Connell, M. (2006). Are two heads better than one?: Object-focused work in physical and in virtual environments. Proceedings of the ACM Symposium on Virtual Reality Software and Technology, 287-296.

[32] Butcher, P., Ritsos, P. (2017). Building Immersive Data Visualizations for the Web. 2017 International Conference on Cyberworlds (CW), 142-145.

[33] Cordeil, M., Dwyer, T., Klein, K., Laha, B., Marriott, K., Thomas, B. (2017). Immersive Collaborative Analysis of Network Connectivity: CAVE-style or Head-Mounted Display? IEEE Transactions on Visualization and Computer Graphics, 23(1), 441-450.

[34] Yang, Y., Jenny, B., Dwyer, T., Marriott, K., Chen, H., Cordeil, M. (2018). Maps and Globes in Virtual Reality. Computer Graphics Forum, 37(3), 427-438.

[35] Yalong Yang, T., Dwyer, Jenny, Marriott, Cordeil, Haohui Chen. (2019).

Origin-Destination Flow Maps in Immersive Environments. IEEE Transactions on Visualization and Computer Graphics, 25(1), 693-703.

[36] Hurter, C., Riche, N., Drucker, S., Cordeil, M., Alligier, R., Vuillemot, R. (2019). FiberClay: Sculpting Three Dimensional Trajectories to Reveal Structural Insights. IEEE Transactions on Visualization and Computer Graphics, 25(1), 704-714.

[37] Maxime Cordeil, Andrew Cunningham, Tim Dwyer, Bruce H. Thomas, and Kim Marriott. 2017. ImAxes: Immersive Axes as Embodied Affordances for Interactive Multivariate Data Visualisation. In Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17). ACM, New York, NY, USA, 71-83. DOI: https://doi.org/10.1145/3126594.3126613

[38] Ferrand, G., English, J., Irani, P. (2016). 3D visualization of astronomy data cubes using immersive displays.

[39] Oh-Hyun Kwon, C., Muelder, Kyungwon Lee, Kwan-Liu Ma. (2016). A Study of Layout, Rendering, and Interaction Methods for Immersive Graph Visualization. IEEE Transactions on Visualization and Computer Graphics, 22(7), 1802-1815.

[40] Azzag, H., Picarougne, F., Guinot, C., Venturini, G., Lina, C. (2006). VRMiner: A Tool for Multimedia Database Mining with Virtual Reality. In Processing and Managing Complex Data for Decision Support (pp. 318–339). Idea Group.

[41] Bach, B., Sicat, R., Beyer, J., Cordeil, M., Pfister, H. (2018). The Hologram in My Hand: How Effective is Interactive Exploration of 3D Visualizations in Immersive Tangible Augmented Reality? IEEE Transactions on Visualization and Computer Graphics, 24(1), 457-467.

[42] Charnine, M., Kuznetsov, K., Zolotarev, O. (2018). Multilingual Semantic Cyberspace of Scientific Papers Based on WebVR Technology. 2018 International Conference on Cyberworlds (CW), 435-438.

[43] Chun, W., Lei, L., Jiwu, Y., Liangyun, Z., Gang, L. (2011). Development of Emergency Drills System for Petrochemical Plants Based on WebVR. Procedia Environmental Sciences, 10(PA), 313-318.

[44] Millais, P; Jones, SL; Kelly, R, Exploring data in virtual reality: Comparisons with 2d data visualizations, Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, 2018, 2018-April.

[45] Hube, N., Müller, M. (2018). The Data in Your Hands: Exploring Novel Interaction Techniques and Data Visualization Approaches for

Immersive Data Analytics. VisBIA@AVI.

[46] B. Carrion, T. Onorati, P. Diaz and V. Triga, ”A taxonomy generation tool for semantic visual analysis of large corpus of documents”, 2019.

[47] Teresa Onorati, Paloma Diaz, Telmo Zarraonandia, Ignacio Aedo: Designing an Immersive Visualization for Semantic Datasets. CHI’19 Extended Abstracts, May 4-9, 2019, Glasgow, Scotland, UK

[48] Das, Kaushik ; Borst, Christoph W (2010). An evaluation of menu properties and pointing techniques in a projection-based VR environment. 2010 IEEE Symposium on 3D User Interfaces (3DUI), 47-50

[49] Jan Ehrlich, ”Exploring Mid-air Interaction for Menu Control in Virtual Reality with regard to Task Performance and User’s Preference”, 2017

[50] J. Brooke, ”SUS - A quick and dirty usability scale”, Usability evaluation in industry, 1996.

[51] Lund, Arnold. (2001). Measuring Usability with the USE Questionnaire. Usability and User Experience Newsletter of the STC Usability SIG. 8.

[52] Roy S. Kalawsky, VRUSE — a computerised diagnostic tool: for usability evaluation of virtual/synthetic environment systems, Applied Ergonomics, Volume 30, Issue 1, 1999, Pages 11-25, https://doi.org/10.1016/S0003-6870(98)00047-7.

[53] Kennedy, R.S., N.E. Lane, K.S. Berbaum, M.G. Lilienthal, Simulator Sickness Questionnaire: An Enhanced Method of Quantifying Simulator Sickness. The International Journal of Aviation Psychology, 1993. p. 203-220.

[54] Bob G. Witmer, Michael J. Singer. Measuring Presence in Virtual Environments: A Presence Questionnaire. Presence: Teleoperators and Virtual Environments, 1993. p. 225–240. https://doi.org/10.1162/105474698565686