Masaryk University
Faculty of Informatics

# Multimodal Virtual Mind Map for Future Workplace

Master's Thesis

**Bc. David Kuťák**

Brno, Spring 2019

Masaryk University
Faculty of Informatics

# Multimodal Virtual Mind Map for Future Workplace

Master's Thesis

Bc. David Kuťák

Brno, Spring 2019

*This is where a copy of the official signed thesis assignment and a copy of the Statement of an Author is located in the printed version of the document.*

# Declaration

Hereby I declare that this paper is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Bc. David Kuťák

**Advisor:** doc. Fotis Liarokapis, PhD

## Acknowledgements

First, I would like to express my gratitude to my supervisor doc. Fotis Liarokapis, PhD for his guidance, continual support, and valuable feedback. I would also like to thank all the people from the Human-Computer Interaction Laboratory, especially BK, MD, and FŠ, and everyone who helped me to find people for the user study, particularly LM, MM, MP, and AP. Special thanks also go to my family, friends, my tarantula, and Casa del Tridente. Since there are some "Thank you" items[1] left in my inventory, if you feel like you deserve one, feel free to take it.

1. Health + 1, Happiness + 1, Depression - 5 | disposable | cannot be sold to merchants

# Abstract

This thesis presents a collaborative multimodal virtual reality mind map application allowing several participants to experience the process of brainstorming. The mind map is used as a tool to organize generated ideas. Multimodality is achieved by using both the controller and speech to text technology as a means of input. The results of a performed user study with 32 participants are summarized and discussed. The system performs according to the specifications. Its most substantial advantage is immersion, while the most significant drawback lies in an unconvincing quality of speech recognition.

# Keywords

virtual reality, VR, virtual environment, collaboration, collaborative virtual environments, mind maps, brainstorming, speech recognition, multimodality, HTC Vive, Unity

# Contents

# List of Tables

# List of Figures

# 1 Introduction

Since the year 2014, virtual reality is becoming more popular and is gaining its ground. [1] People in numerous industries are trying to use virtual reality for various purposes, ranging from car design [2] to military. [3] There are tens of manufacturers developing virtual reality headsets [4] and the market is expected to grow significantly in the following years. [5] Moreover, with the stable increase in the number of people connected to the internet [6] and connection speeds, [7] the virtual reality is also used via a network for multi-user experiences, for example, games [8] and conferencing. [9] Such types of applications are also known under name collaborative virtual environments, and the work presented in this thesis falls under this category as well.

The goal of this thesis is to create a mind mapping application which can be used for remote brainstorming, i.e., an activity aimed at generating lots of ideas related to the given problem. When groups of people who are far away from each other want to brainstorm, there are two solutions in general. They can use some common application, like Skype, which makes it possible to connect all of them by using voice or video chat. However, this approach usually suffers from confusing communication, when there are more people present, and lack of features aimed at brainstorming. It is also far away from feeling realistic, i.e., like an on-site meeting. The second solution is that one group of people travels to the location of the other one and they run a classic brainstorming session with the help of post-it notes. The first problem with this solution are the costs related to travelling. Besides that, reshuffling of post-it notes is a difficult task because notes often fall from the wall and this problem is becoming worse every time a note is moved. Another problem occurs when trying to move several notes at once to another position on the wall. This process is rather slow and cumbersome. Mapping relationships between post-it notes is another difficult task as one needs to draw lines among them, which will need to be redrawn every time the notes move. Also, the text written on the notes might be hardly readable for some participants, especially when viewed from a bigger distance. Finally, post-exercise analysis is difficult; it typically involves a photograph of the result and then manual transcription into a different format like Word document or mind map in some conventional software. The system presented in this thesis tries to offer a solution to these problems.

The application was developed as a part of the cooperation between Konica Minolta Laboratory Europe and Masaryk University. Parts of this thesis, including the results of the user study, are going to be published as a paper. Since the paper is still in review, it can not be cited at the time of writing.

## 1.1 Aims And Objectives

This thesis aims to develop a collaborative multimodal virtual reality application which will utilize mind maps as a brainstorming tool. The objectives are:

- Design and create a multi-user virtual environment suitable for given purpose.
- Enable users to cooperate on the process of mind map creation. The mind map should support
  - creation and deletion of nodes;
  - node movement;
  - visualization of relationships between nodes; and
  - modification of additional nodes' properties like label or color.
- Implement mind map export to some common format.
- Implement support of voice recognition for a description of nodes' labels. The application will be used by English speaking users, and no other languages have to be supported.
- Perform a user study with at least 30 participants and evaluate the results.
- The application should work with virtual reality headset HTC Vive (Pro).

These are the desired goals determining both the functional and non-functional requirements of the application. The latter category also includes requirements posed by hardware and its limitations. These involve, for example, the ability to render the application at 90 frames per second on virtual reality enabled computers.

## 1.2 Thesis Structure

The contents of the thesis are structured as follows. Chapter 2 introduces the topics related to this thesis, provides some background information and discusses the related work. Chapter 3 presents the system from the design and architectural point of view and describes the required hardware. Chapter 4 dives into the implementation details of the application including game engine, networking, and individual operations with the mind map. This chapter also describes the controls scheme of the system. Chapter 5 reveals the outputs of the user study and examines the findings. The final chapter summarizes the thesis, discusses the most important outputs of the user study and presents ideas for future development.

# 2 Background

## 2.1 Mind Maps

During the history of human civilization, many different ways of visual mapping were explored [10], some of them closely resembling mind maps as we know them today. In its core, mind maps serve as an effective tool to organize information visually [11] and can be described as diagrams consisting of nodes and lines between them. They can take many forms, from simple drawings on a piece of paper to complex, colorful maps with hundreds of nodes created in specialized software. Example mind map is shown in Figure 2.1.

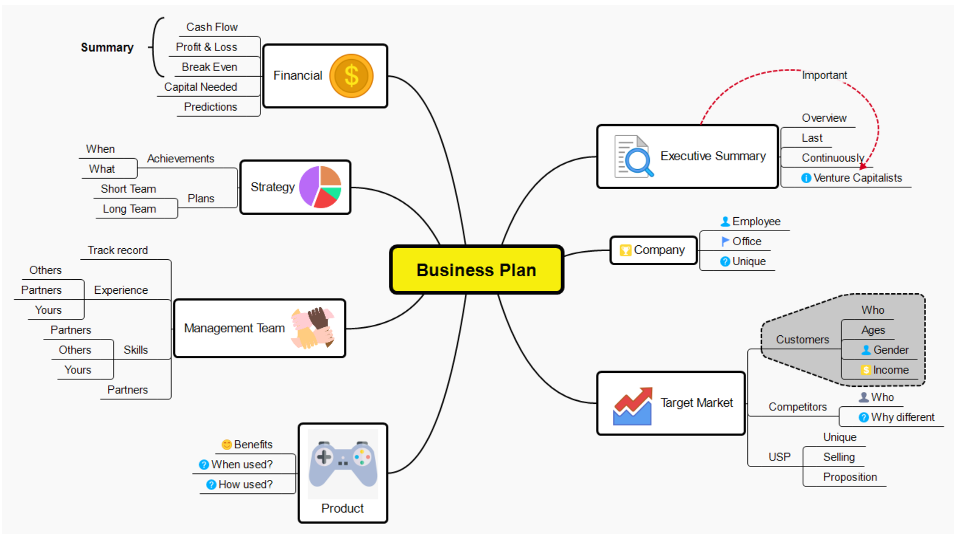


Figure 2.1: Example mind map created in XMind software [12]

The term "mind map" itself was first introduced by the British author of popular psychology and TV host, Tony Buzan. He described the idea of mind mapping together with some suggested guidelines in many of his books with most important ones being Use Your Head [13] and The Mind Map Book: How to Use Radiant Thinking to Maximize Your Brain's Untapped Potential [14]. Amongst Tony Buzan's suggestions, the key ones are the presence of one central topic and heavy utilization of colors and images together with simple, straightforward keywords and clear associations. He summarized seven steps to making a mind map in his book The Ultimate Book of Mind Maps: [15]

1. Start in the center of a blank page turned sideways.
2. Use an image or picture for your central idea.
3. Use colors throughout.
4. Connect your main branches to the central image and connect your second- and third-level branches to the first and second levels and so on.
5. Make your branches curved rather than straight-lined.
6. Use one key word per line.
7. Use images throughout.

Regarding the sixth step, it is worth mentioning that there are two slightly distinct approaches to mind mapping. The method presented by Tony Buzan appends labels to the lines and thus basically removes the nodes; the result is therefore visually resembling a real tree. The second approach, also used in this thesis, labels the nodes instead and results in maps looking like undirected graphs. What technique is chosen is just a matter of preference. Since mind maps can take many forms and are created with different goals, guidelines mentioned above can be shaped according to the specific usage. However, despite the possible differences in their form and purpose, all mind maps share some common characteristics. For example, when compared to the linear form of note taking, they have several advantages as cited in [13, p. 91-92]:

- The center or main idea is more clearly defined.
- The relative importance of each idea is clearly indicated. More important ideas will be nearer the center, and less important ideas will be near the edge.

- The links between the key concepts will be immediately recognizable because of their proximity and connection.
- As a result of the above, recall and review will be both more effective and more rapid.
- The nature of the structure allows for the easy addition of new information without messy scratching out or squeezing in, and so forth.
- Each map made will look and be different from each other map. This will aid recall.
- In the more creative areas of note making such as essay preparations, the open-ended nature of the map will enable the brain to make new connections far more readily.

Mind maps have many applications thanks to their visual nature and immense possibilities. Students can use them for note taking as they can concisely capture significant amounts of content together with appropriate relations between stored information. [16] [17] Their usage was proven beneficial also in the field of analysis of complex data. [18] [19] Mind mapping can also be used to enhance essay writing skills. [20] Finally, this technique turned out to be well usable during brainstorming sessions. [21] [22]

## 2.2 Brainstorming

In 1942, advertising executive Alex Faickney Osborn introduced the idea of creative group activity called "brainstorming" in his book How To Think Up. [23] It is one of the ideation techniques aimed at generating lots of ideas related to the given problem. Such an exercise can have many forms, and a variety of tools can be used to increase the effectiveness and quality of the process. Although it is usually a group activity, so-called "individual brainstorming" is performed as well. In general, the goal of brainstorming is not to provide solutions for complex problems, but to deliver as many ideas as possible serving as a reasonable basis for later decisions. [24]

Even though the ways the brainstorming is held may differ, there are some recommended rules to follow to get the most of the process. One of the participants is usually appointed a leader who makes sure that the exercise is going in the right direction. Leader also ensures the "flow" of the session by providing some suggestions when it seems that the actors are stuck and unable to come up with new ideas. Since brainstorming is all about generating ideas, it is necessary to store them somewhere. Whatever technique is chosen, it is important to make sure that everyone can see everything and that the ideas are stored in an anonymous form. [24]

James L. Adams summarized Osborn's four main rules of brainstorming in his book Conceptual blockbusting: A guide to better ideas, and emphasized the importance of these guidelines for successful brainstorming: [25]

- **No evaluation is permitted**
  - Criticism or other types of judgment will negatively influence the effectiveness of the process. People might become afraid to share their ideas, or they might spend time defending them instead of creating new ones.
- **Wild ideas are encouraged**
  - Even the wildest ideas should be shared. There is no bad idea. Making up new ideas is more complicated than taming down existing ones.
- **Quantity leads to quality**
  - Having lots of ideas means having a lot of useful thoughts on the subject which can lead to a good decision later on.
- **Build on ideas of others**
  - Creation of new ideas based on already existing ones tends to lead to ideas which are superior to already existing ones and which might provide a different perspective.

Adams considers the first rule to be the most important one and crucial for productive brainstorming. This rule is also a reason why ideas are stored without the author's identification as mentioned earlier. If the author's name was present, participants might have been tempted to talk to their authors, or they could feel an urge to compare the quality of ideas generated by participants. Such behavior would negatively influence the quality of the brainstorming.

## 2.3 Virtual Reality

### 2.3.1 The Definition of Virtual Reality

The term virtual reality (hereafter VR) itself has many definitions. Cambridge English dictionary describes it as "a set of images and sounds, produced by a computer, that seem to represent a place or a situation that a person can take part in". [26] Jason Jerald points out in his book The VR Book: Human-Centered Design for Virtual Reality an interesting fact that this term actually turns out to be an oxymoron when meanings of words "virtual" and "reality" are analyzed separately. [27, p. 9] In any case, for this thesis, the VR will be defined as a computer-generated digital environment which is highly immersive for the user for whom the experiences and interactions, happening inside the virtual world, seem to be believable and consistent with its rules. Specifically, the immersion, an ability to convince the user that he or she is present in the virtual environment, is essential. On the other hand, although it might seem that the ultimate goal of VR is to become as highly immersive as possible, it was shown that for certain types of tasks the highly immersive systems might provide negligible advantages in comparison to less immersive ones. [28] Furthermore, immersion might also be a negative experience in specific scenarios. [29]

It is good to point out that previously stated definitions of the VR do not talk about any specific devices used to present the VR to the user. Nowadays, the term VR is mostly used in association with VR headsets, discussed in later sections of this thesis, which are however only one element of the set of possible devices suitable for virtual reality portrayal.


Figure 2.2: Virtual reality parachute training simulator [30]

### 2.3.2 The Applications of Virtual Reality

Virtual reality can be used in many different areas, beginning with the entertainment industry which turned out to be a big market for this type of technology as VR headsets are used to play games [31] and watch movies [32]. Sony PlayStation VR, headset developed exclusively for the Sony PlayStation gaming console, has sold more than 1.7 million units in the year 2017 becoming the best selling VR device of the year. [33] Another large field for the VR is military. They use VR for example as a part of gunship crew trainer technology, [34] virtual parachute trainer [30] (Figure 2.2) and naval simulator [35]. Additional use cases include improved battlefield visualization and simulation of scenarios based on interaction with computer-driven human characters. [36] VR is also used for education. Existing research confirms improvement in the performance of different medical groups when VR is used, [37] [38] and several other study fields are taking advantage of VR as well. [39] [40] VR furthermore turned out to be useful in the building industry [41] and can also serve as a useful tool for remote collaboration. [42]

### 2.3.3 The History of Virtual Reality

Although the VR seems to be a relatively new topic, as it is most popular in the last few years, [43] the reality is that this subject has been already discussed for a long time. As one of the first predecessors of the VR devices can be considered the stereoscope invented in 1838 by Charles Wheatstone. [44] Although this device is not providing a full-fledged virtual reality, as described in definitions in section 2.3.1, the design of the invention and the idea of using stereoscopic pair of images to achieve three-dimensional feeling, are the essential elements of modern VR devices.

Probably the first device offering multi-sensory virtual reality experience was The Sensorama Machine patented in 1962 by Morton Heilig. [45] It offered a 3D screen with binaural sound, seat vibrations, and wind and scent generators. The experience provided by this machine was not interactive as it consisted only of several prerecorded movies. Sensorama can be seen as a predecessor of recent 4DX motion picture technology. [46]

In the year 1968, Ivan Sutherland, together with his student Bob Sproull, created the first head mounted display (hereafter HMD) named The Sword of Damocles (see Figure 2.3). [47] This HMD was connected to the computer, instead of the camera, and enabled users to view simple wireframes with perspective changing according to the head movements. The negative side of this HMD was its weight. It was so heavy that it was suspended from the ceiling. This also explains the name of the device.

Although there have been some inventions in this field, it was not until 1987 that the term "virtual reality" was popularized by Jaron Lanier, founder of the VPL research. [44] This company was the first one to sell VR goggles and gloves. In the 90s, experiments with VR continued and two major gaming companies, Sega and Nintendo, tried to release their gaming VR HMDs to the public. The product of Sega failed already in the prototype stage due to the technical and financial difficulties. Nintendo succeeded and started selling the headset, but the sales figures were rather bad. In the end, both products ended up as a commercial failure. [49] There were also several experiments to present virtual reality without usage of HMDs. Cave Automatic Virtual Environment (hereafter CAVE) from 1992 being one of the

Figure 2.3: The Sword of Damocles HMD in action [48]

most successful examples. [45] CAVE consists of a box with the user standing inside it, whereas the VR environment is stereoscopically projected upon the walls. The screen is projected usually at least onto four of the six sides with more expensive CAVE systems being able to project screen upon all walls of the box. When compared to HMDs, CAVE has both benefits and drawbacks, what prevails depends on the exact scenario. [50]

Thanks to the rapidly growing performance of computers and recently also smartphones, the 21st century allowed VR to spread in popularity. This is most significant in the last few years. Since the year 2014, when the Oculus company was acquired by Facebook, VR HMDs are given much more attention. [49] Nowadays, there are three main categories of VR headsets: [51]

- Tethered VR HMDs
- Mobile VR HMDs
- Standalone VR HMDs

The first two categories require computer, or smartphone, resp., to work. The last type represents headsets which can operate independently without other devices. Tethered headsets usually offer the best screen quality and performance; they, however, tend to be more expensive and more complex in terms of setting up. Figure 2.4 shows three, currently best-selling, tethered HMDs [52] – PlayStation VR, Oculus Rift, and HTC Vive.


Figure 2.4: Left to right: PlayStation VR, Oculus Rift, HTC Vive [53]

Regarding the future of the VR, it was removed from the 2018 Gartner's hype cycle due to being "almost mature". [54] Therefore it seems that this type of experience reached stability and is considered usable. Gartner's hype cycle, on the other hand, includes Mixed reality and Augmented reality, two emerging technologies having some similarities with the VR. The time will show what kind of experience will be the dominant one.

## 2.4 Multimodal Systems

The term "multimodal system" consists of two words. For the former one, slightly different definitions exist, depending on the specific context. Cambridge English dictionary describes the term "modality" as "a particular way of doing or experiencing something". [55] As the prefix "multi" suggests, multimodality can be seen as several distinct ways of doing a certain action. The word "system" hints that the multimodality will be in this case related to the features or interface of some system. Summing it up, multimodal systems can accept user input in more than one modality. They are, for example, able to process voice or gesture input on top of traditional ways like keyboard, mouse or controller. [56]

The idea of multimodal systems and interfaces is based on a human - human interaction. [56] During communication, humans utilize most of their five senses – sight, hearing, touch, smell, and taste. Even during a casual conversation, at least sight, hearing and smell are all at the same time actively taking part in the interaction, and each of these senses can be considered a modality as defined above.

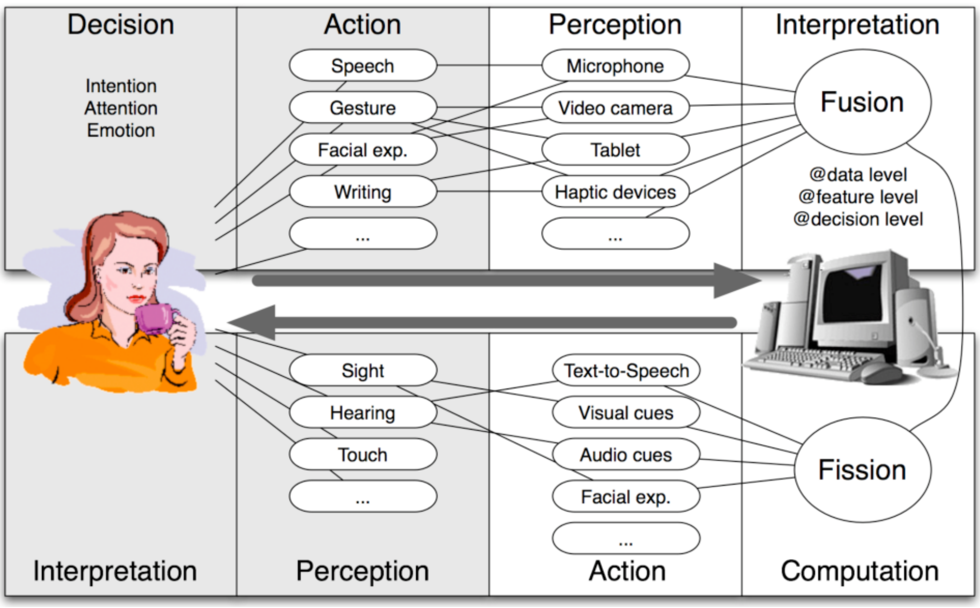


Figure 2.5: Multimodal man-machine interaction loop [57]

Another term related to the multimodal interaction, present also in the loop in Figure 2.5, is fusion. [58] Fusion mixes several input or output modalities and interprets them as one expression. A good example of fusion is Bolt's Put-that-there system [59] using gesture and speech together to move given object to the position pointed to by a finger. [60, p. 419] In the book Readings in Intelligent User Interfaces [61], several benefits of multimodal interfaces are summarized:

- Efficiency
  - The most suitable modality is used for each task.
- Redundancy
  - The probability that the interaction will be smooth is increased thanks to more available modalities suitable for achieving the given goal.
- Perceptability
  - Increased when the tasks are facilitated in spatial context.
- Naturalness
  - Thanks to the presence of more modalities, the user can choose the most convenient one which makes the interaction more natural.
- Accuracy
  - Increased when there exists a modality which can more precisely describe the desired goal than the main modality.
- Synergy
  - Occurs when one modality can help to correct, improve or influence in any other way ambiguities from another means of interaction.

However, it is good to realize that multimodal interfaces have disadvantages as well. [62] Some of them are related to the increased development complexity. Multimodal systems are getting harder to test and develop with every new supported modality. [63, p. 7-13] It is necessary to make sure not just that all modalities work well as individual components but also that their combinations are working as expected. Implementing such interfaces might also require interdisciplinary knowledge – some modalities might be based on image processing [64], some on speech recognition [65] and even brain-computer interfaces (hereafter BCI) can be utilized. [66] Another danger of multimodal systems is overcomplexity. While having more modalities might make the interaction more efficient and natural, it might also happen that the user will spend too much time thinking about the best modality to use and end up feeling intimidated by the number of possibilities. Finally, even if the application interface is well designed, it will always hold that each new modality will cost some time when explaining the full extent of the system to the user.

Besides the famous Bolt's Put-that-there system, there are several examples of more recent multimodal interfaces. Some researchers designed multimodal systems suitable for elderly people as they realized that common application interfaces might not be well usable for such users. [67] Others tried to create virtual keyboards designed specifically for older people. [68] Multimodal interfaces were also used to help disabled people control robotic wheelchairs [69] and interact with computers. [70] There were also experiments with usage of multimodal interfaces inside systems assisting doctors during the surgery. [71] Scientists from Hasselt University proposed a collaborative virtual brainstorming environment based on a shared virtual workbench. [72] Their systems used mix of pointing device, gestures, menus and camera interface to help users to work together on a drawing.

## 2.5 Collaborative And Mind Mapping Software

There are several software alternatives for the mind map creation with XMind [12] and iMindMap [73] being the most famous ones. Although these solutions offer wide range of features, they provide only traditional desktop and mobile interfaces without collaboration possibilities. Another popular tool, MindMeister [74], enables users to collaborate in the real-time, however it offers only web and mobile interface. In the world of VR-enabled applications, Noda [75] is one of the most progressive alternatives. It does not, however, offer collaboration possibilities. Another difference is that the Noda utilizes spatial mind maps with nodes being positioned anywhere in the three-dimensional space, whereas the application presented in this thesis sticks to the traditional two-dimensional mind maps. Having a three-dimensional mind map presents some advantages like increased ability to employ spatial thinking and theoretically infinite place for storing ideas. On the other hand, spatial placement might decrease the clarity of the mind map as some nodes might be hidden behind the user or other nodes. The one-to-one correspondence with traditional mind mapping software is lost as well which makes it hard to export the results for later processing and review. This would decrease the usability of the outputs created inside the virtual reality, and it is the reason why the software presented in this thesis works with two-dimensional mind maps. Another VR mind mapping tool is Mind Map VR. [76] Authors of this application decided to adopt similar approach as the Noda and offer spatial mind maps without collaboration possibilities. Interesting feature of the Mind Map VR is the ability to switch VR environments. Regarding the Collaborative Virtual Environments (hereafter CVE), that is VR environments shared by multiple users connected via computer network, rumii [9] and MMVR[1] [77] are to certain extent similar to the work presented here. Rumii is aimed mostly at VR conferencing and presentations and does not offer mind mapping functionality. It does, however, share business-oriented approach and some functions like PDF[2] viewer. Rumii is also a source of inspiration for future development since it offers more sophisticated avatars and communication toolset, large VR environment and some features which could help to improve the effectivity of the brainstorming. MMVR is collaborative mind mapping tool controlled with hand gestures. Similarly to Noda, it also utilizes three dimensional placement of nodes. It is aimed purely at mind maps so it lacks some brainstorming functionalities, like voting about ideas, in comparison to system presented in this thesis. MMVR does, however, offer some additional features like possibility to insert pictures into the mind map.

1. MMVR is full name of the system, not a shortcut
2. PDF – Portable Document Format

# 3 System Design And Architecture

This chapter describes the core components of the system, its functional and non-functional requirements and supported hardware. It is worth mentioning that descriptions in following sections of the text are related to the latest version of the system presented in this thesis whereas the corresponding user study was executed with an older, less evolved version. Some of the main differences between these two versions are discussed in relevant parts of the text.

## 3.1 Architecture Diagram

The purpose of the Architecture diagram in Figure 3.1 is to show the high-level structure of the system, i.e., mainly its components and basic relations between them. At the lowest level, there are three core parts of the system – software describing the functionality through the code, hardware executing the code and in the form of peripherals, and users interacting with the final application. The software part can be split into two main categories, namely the VR environment and Services, according to the type of functionality the individual elements deliver. VR environment consists of objects which can be seen and possibly offer some interaction; it is, therefore, the front end part of the application. Examples of such objects are user avatars, mind map canvas or all 3D models in the environment. The second category, Services, is a set of mainly back end features which are delivering functionality which is useful or even necessary but not directly visible. Services include networking, speech to text support or logging of user behavior.

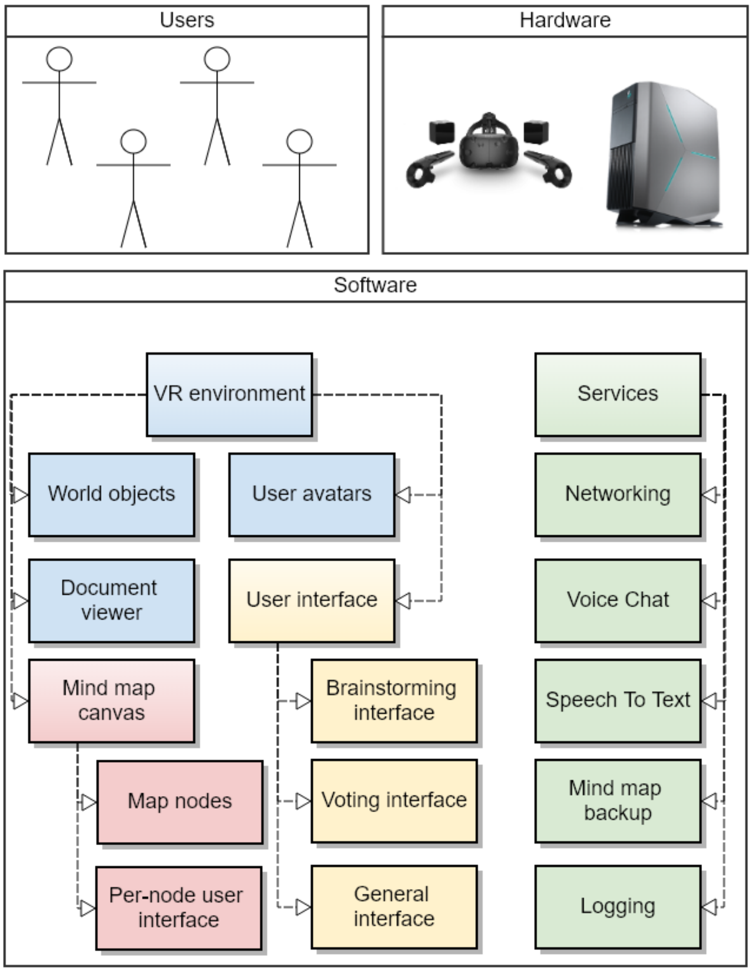


Figure 3.1: Architecture Diagram of the system

## 3.2 Hardware

The system is designed to be used with the virtual reality HMD HTC Vive (including the Pro version). This headset has a refresh rate of 90Hz and supports room scale experience. The refresh rate of the headset poses a requirement on the application performance – VR-enabled computers should be able to run the application at 90 frames per second or faster to achieve the best experience. One HTC Vive Controller is used as an input device interacting with the application. Figure 3.2 shows the description of this controller.


Figure 3.2: Description of the HTC Vive Controller buttons (controller's image from [78])

Although the system is primarily developed for the HTC Vive, it also works without any headset and users can interact with the application using traditional mouse and keyboard. However, the interface and controls are not adapted to this means of interaction, so although the functionality is delivered, it is not a smooth experience. The application also works with the HMD Oculus Rift but cannot be properly controlled because the corresponding controller is missing some buttons.

There were also experiments with Leap Motion hand tracking device [79] and support for this means of interaction was added as well so, in theory, the HTC Vive controller can be replaced with this technology in order to control the system with bare hands. The desired use case of this device was to point with the index finger at elements with which the user wishes to interact. However, in practice, the Leap Motion turned out to be unsuitable for this purpose due to the gesture detection quality being insufficient and unreliable in such conditions. Also, the field of view of the device was a limiting factor. To explain in more detail, when the Leap Motion was mounted to a HMD, it had problems with the gesture mentioned above for two reasons. First, sometimes it was physically impossible to see the gesture from the point of view of the device. Although the user knew that he is pointing with an index finger, the finger itself was hidden by the back of the hand when viewed from the position of the Leap Motion. The second problem is related to the "shaky behavior" of the detection. Even when the gesture is correctly detected, the index finger's computer representation suffers from noticeable "shaking" which significantly decreases the precision of the pointing. Further experiments with the device revealed that it works best when the palm is facing the head, all fingers are well visible and the hand is rather close to the device and entirely in its field of view. When most of these conditions are fulfilled, the device provides acceptable results. The way the hand gesture controls should be used in this application does not meet these conditions, and the detection quality is therefore insufficient. Generally said, at the moment, it is nearly impossible to effectively interact with the system using Leap Motion hand tracking. This feature is therefore currently considered as a dead end and explained no more in this thesis. In the future, the possibility to use more Leap Motion units, or similar devices, at the same time to increase the detection quality will be examined.

## 3.3 Use Case Diagram

UML[1] Use Case diagram in Figure 3.3 shows users of the system together with corresponding core use cases. These determine what interactions will happen between the user and the system and therefore also functionality the system should offer. As mentioned in section 2.2, members of the brainstorming group are usually one moderator and several participants. In case of this system, it is considered that the host of the networking session, i.e., the user who starts the server to which other users can connect, is automatically also the moderator of the brainstorming. There are therefore two basic types of users from the system point of view. Regarding the individual use cases, they include basic actions like modification of mind map and possibility to take part in voting.

1. UML – Unified Modeling Language

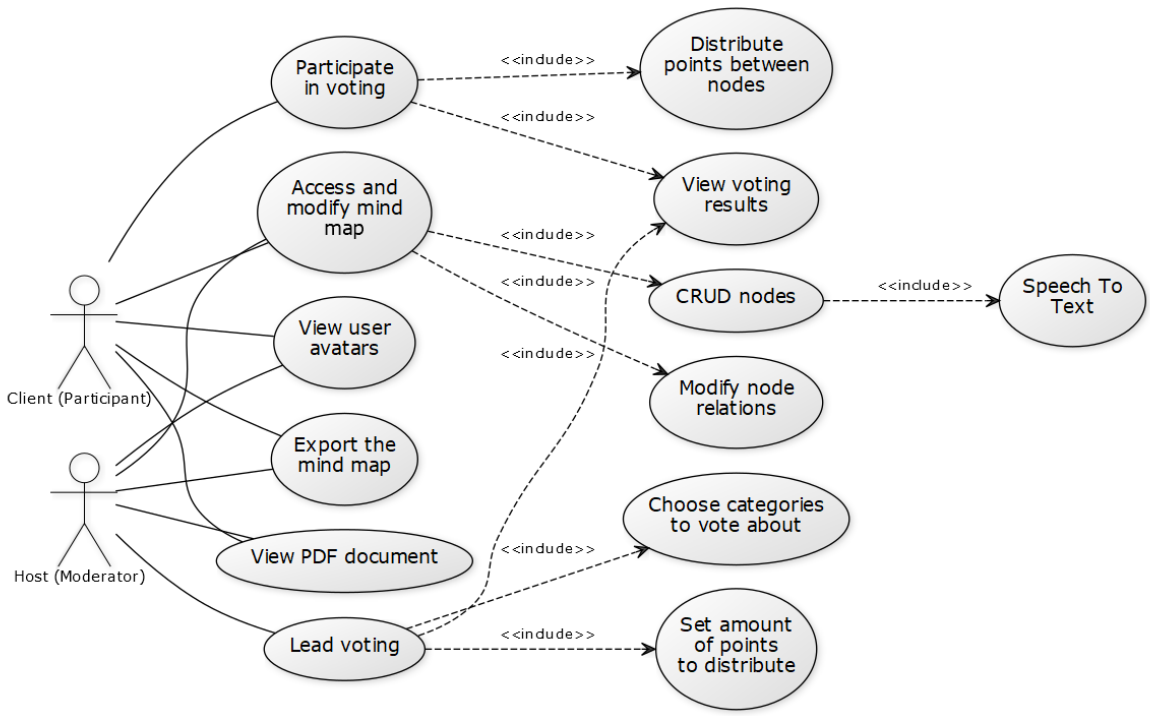


Figure 3.3: System Use Case Diagram

## 3.4 Multimodality

The multimodality is achieved via combination of the controller (or possibly computer mouse) and speech to text technology. The controller is used to interact with system components and to perform tasks related to the organization of the mind map, for example, selection of nodes and reshuffling. Moreover, simple gesture-based actions are implemented to simplify the workflow and to make the system more natural. They are used for multiple selection (section 4.6.2) and node creation (section 4.6.1). Speech to text technology is utilized as an input tool for node labels and possibly also for simple voice commands.

The most commonly occurring fusion between those two input modalities is the following: when the user initiates the voice recording (controller modality), he or she says something (speech input modality) and the system changes the label of the selected node (controller modality) to the recognized text. In the case of voice commands, the controller serves again as a tool determining "what" should be changed, whereas the speech input determines "how".

## 3.5 Environment

An environment is a crucial part of all systems which can be categorized as collaborative virtual environments. It does determine not only the system visuals but also placement of the individual interactive components and therefore significantly influences both the immersivity and usability. Since the application was designed to stick to traditional two-dimensional mind maps, it was clear that the environment must offer some wall, or more specifically, canvas, which will serve as a place where the mind map will be visualized. A space for the user interface must accompany this canvas. These are the essential things from the functionality perspective. Besides that, to support the immersivity, the environment must be somehow interesting. On the other hand, it should not be disturbing, and therefore the inclusion of interactive things just for the sake of realism is undesirable. Such are the design basics for the virtual environment, which is in more detail described in section 4.5.

# 4 Implementation

This chapter presents all features of the application and provides an in-depth overview of their implementation.

## 4.1 Game Engine

Although the presented system is not a game, it was developed in a game engine Unity, [80] specifically version 2017.4.24f1 LTS [1]. Even though there are newer versions, this is the last LTS one providing full support for utilized networking technology.

Unity Engine is aimed at game developers but due to its huge popularity and wide range of features, it is a good choice for development of virtual reality applications. 60% of worldwide AR[2]/VR content is made with Unity. [81] The program is written in C# language.

## 4.2 Virtual Reality Support

Development of interactive virtual reality applications in Unity is possible mainly thanks to the support of SteamVR SDK[3]. [82] This plugin, developed by Valve, provides a layer of abstraction allowing applications to interact with given VR device without a necessity to distinguish between specific hardware. In addition to SteamVR, popular library, called Virtual Reality Toolkit (abbreviated to VRTK), [83] was used as well. VRTK provides some additional features for more convenient VR development in Unity like an interaction with objects and user interface, support for virtual space locomotion and rigid body physics.

## 4.3 Interaction

As already stated in section 3.2, one HTC Vive controller is used to interact with the application. Although some actions can be possibly

1. LTS – Long-Term Support; a version of Unity with two years long official support
2. AR – augmented reality
3. SDK – Software development kit

| Version/state | Active | Selecting |
|---|---|---|
| **User study** | Trackpad (touchpad) press | Trigger press |
| **Latest** | Trackpad touch | Trackpad press |

Table 4.1: HTC Vive controller's button mappings used in the application

performed with voice, controller is still the necessary and main input tool. It is mirrored into the VR application and has an ability to emit a laser pointer there determining the aim direction. In general, there are three states in which the VR controller can be:

- Inactive
  - The controller is visible but does not emit a laser pointer
- Active
  - The controller emits a laser pointer which can hit game objects. In this state, pointer can trigger "hover" events, i.e., acts similarly as a mouse cursor when no button is pressed.
- Selecting
  - When the controller is in this state, the emitted laser pointer can be used to press buttons hit by its ray and to trigger similar actions.

During the development of the application, two different button mappings were used with the current one being simplification of the initial version. Table 4.1 shows what buttons are used in each version to switch controller's states.

### 4.3.1 Speech To Text

Text input is a big challenge for all virtual reality applications as the traditional keyboard cannot be comfortably used due to not being visible. Besides that, it also disallows users to move freely around. The most straightforward idea is to use a virtual keyboard, but this approach is not very effective and certainly slower than the traditional keyboard – especially with only one controller as in case of this application. It was therefore decided to use speech to text technology as the means of input. Moreover, the ultimate goal was to find both usable and freely available speech to text solution; the list of available options was thus quite reduced.

First, there were several experiments with CMU Sphinx library, [84] which is a relatively well-known offline solution for speech recognition. It is written in C language, so it was necessary to compile a DLL[4] and create appropriate C# wrappers to make this library work with Unity in a demo application. The tests revealed that the CMU Sphinx is almost unusable for our purpose in its initial state. It would require significant tweaking – mainly a good dictionary, the Hidden Markov Model and Language Model – to provide acceptable results. Moreover, the broader the vocabulary of known words is, the more the detection quality decreases. In the end, it was decided that another solution will be examined, namely a cloud-based speech to text service Wit.ai. [85] This company is owned by Facebook and provides free unlimited access to their speech recognition API[5]. Demo application showed that Wit.ai performs quite well and should be enough good for this system. Hence, Wit.ai was chosen as the solution which is currently in use.

Regarding the speech to text implementation in the system, the user uses an appropriate button in the radial menu to start the recording, says the desired input and then ends the recording by pressing the button again. The audio data, captured from the microphone during this period, are saved as Unity's AudioClip data type. The recording is then converted into a WAV[6] audio file uploaded to the Wit.ai servers, which process the audio and return a response containing the detected text. From the perspective of program optimization holds that as soon as the user stops the recording, the audio processing runs on a separate thread to not block the program's execution while waiting for the result.

4. DLL – Dynamic-link library
5. API – Application programming interface
6. WAV – Waveform Audio File Format

### 4.3.2 Voice Commands

Voice commands allow users to execute certain actions just with their voice. First, the details behind the implementation will be examined followed by the explanation why are the voice commands in their current state poorly usable and subject of future improvements.

In its core, voice commands behave similarly as a recording of the node's content. Specifically, the user interface contains button allowing to start and stop the recording of the command. The audio is then processed by Wit.ai service and the returned result is passed to the command parser analyzing the text and executing related actions. Command parser works in two steps. First, it processes the text according to the rules of the grammar, which is defined specifically for the application and has syntax inspired by Backus–Naur form (example in Listing 4.1). This stage includes deciding whether the input is valid and if yes, extracting the information and saving it in a predefined format (data structure ParsingResult). After this step, the parser uses the extracted information to decide what actions to take.

Regarding the grammar rules, it must be non-recursive and unambiguous. Parsing is done in a case-insensitive way and is word-based, i.e., input sequence is space-separated and then processed using a parsing table derived from a grammar. When designing the voice commands, it turned out that each one usually holds up to "three pieces of information". For this reason, ParsingResult data structure, carrying the extracted information, consists mainly of three string fields – Action, Object, Params. Action determines what should be done, Object carries the target to influence and Params contains information about the parameters of the action. Based on these values, appropriate functions are called with correct parameters. To simplify the second step of the parsing, grammar contains special fields, for example "{action ...}" and "{object...}" in Listing 4.1, used to fill the appropriate fields of ParsingResult with unified value, for example "change", although different words, like "update" or "set", can be used to trigger this action from the user's perspective.

As mentioned at the beginning of this section, voice commands are not working as expected. The problem arises from the way speech to text technology works. It tries to create natural sentences; something people would generally say. On the other hand, commands are rather simple and artificial. This difference tends to result in such a behavior that when the user says a command, the voice recognition sometimes returns completely different text, adds some prepositions or replaces some words. It, therefore, tries to create a sentence which seems to be more meaningful. To partially avoid the problem, grammar definition must contain several alternatives for some words to be able to understand, for example, that in a specific context, "three", "tree" and "free" have the same meaning. However, such improvements make the grammar much more complicated. The best solution, which is probably going to replace the current one in the future, is to use some more advanced natural language processing techniques together with making the commands sound more like natural sentences.

Listing 4.1: Excerpt from grammar definition

```
init <command>
<command>     = <change> | <create> | <remove>
<change>      = "change" & <change_obj> |
                "set" & <change_obj> |
                "update" & <change_obj>
                : {action change}
<create>      = "add" & <create_pos> |
                "create" & <create_pos> |
                "at" & <create_pos>
                : {action create_node}
<change_obj>  = <change_node> & <change_node_pos> |
                <change_text> & <change_col>
<change_node> = "node" | "note" | "no" |
                "not" | "know" | "notes"
                : {object node}
(...)
```

## 4.4 Networking

### 4.4.1 Core Functionality

The networking core of the application consists of the Collaborative Virtual Environments platform [86], providing basic CVEs features like visualization and synchronization of user avatars, built on top of the Unity Networking (also UNet) technology. Thanks to UNet, games, and applications made in Unity can deliver multi-user experience via a computer network. Each networked game object has a network identity component determining its unique identification and storing information about its authority. Authority tells whether the object is managed by a server (server authority) or a client (local authority). For non-player objects, default authority is the first one, and it holds that if the client wants to perform some action with the selected object, he or she must have an authority over it. To obtain an authority, the client sends a request to the server, whereas server not just only decides whether to grant the authority or not but also ensures that only one client has authority over the object at the given time. In case of this application, most of the actions are performed via so-called "mind map manager" game object (and accompanying scripts) which is used by clients with authority to modify mind map and execute other related actions. The process of performing an action usually consists of three basic steps happening in the background and therefore hidden from the user:

1. Getting the authority over the object.
2. Executing an action.
3. Releasing the authority.

When a client requests an authority, the request may be denied. Although this might not be a problem in some instances, for example when the user can wait to repeat the action later on, this behavior is problematic in this application. Due to this reason, polling is utilized when trying to gain authority. The three steps mentioned above are split across several game frames which allows waiting during the first step until the authority is successfully obtained. As soon as this happens, the remaining two steps are executed. This guarantees that every network-related function call will be executed and will not fail due to the unsuccessful authority assignment. It is worth mentioning that all actions are almost instant, taking less than a second, so the clients hold the authority over "mind map manager" object only for a short amount of time.

As indicated above, UNet works on a client-server model. More precisely, in this application, one user acts as a server and a client

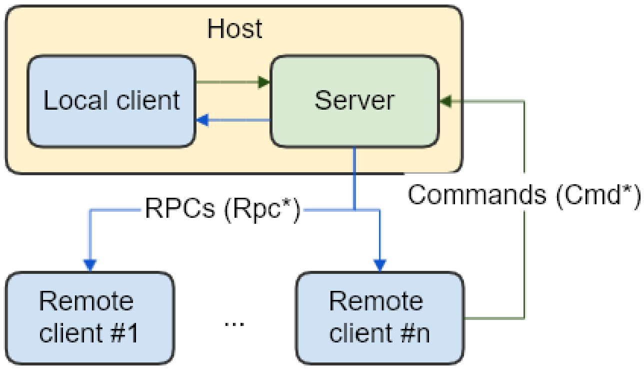


Figure 4.1: UNet function calls

at the same time – he or she is a so-called host – while other users are connected as clients. The Figure 4.1 shows the abstraction of the network session hierarchy and associated function calls. The first thing to mention is that all network-related actions go through the host, or more precisely, server, and it is necessary to make sure they are properly synchronized on all clients. In general, there are two basic types of network function calls to achieve this goal – Commands and RPCs [7]. UNet utilizes C# attributes to add the possibility to label the functions according to their purpose. Commands are functions with attribute "Command" and prefix "Cmd", RPCs have attribute "ClientRpc" or "TargetRpc" and prefix "Rpc", respectively "Target". Commands are called on a client with authority, but they are executed on the server. This means that the appropriate action will influence only the server version of the application (which is shared with local client). RPCs serve for an opposite purpose – they are called on a server but executed on the clients' side – in case of "ClientRpc", all clients execute the function, whereas "TargetRpc" executes the function only on selected client. Both Commands and RPCs have some limitations on the data types of function parameters because passed arguments need to be reliably transferred over the computer network. To illustrate the functionality better, imagine a scenario where a client changes a color of a node. What happens in background is that the code acquires an

7. RPC – Remote procedure call

authority over "mind map manager" object, calls a Command where the arguments are identification of a node and the new color, and then releases the authority. On the server side, the appropriate Command function is actually executed with arguments received from the client. Definition of this function includes calling function *F* locally changing node's color and then calling appropriate "ClientRpc" function containing a call to *F*, again with the original arguments. After this is done, the color of the node is changed on the server and all connected clients and the network session, therefore, stays consistent.

### 4.4.2 Modes Of Connection

In its current state, the application offers four modes of the connection while the desired one can be selected via corresponding user interface elements:

1. Offline
2. Unity Cloud
3. LAN (Local area network)
4. VPN (Virtual private network)

When an Offline option is chosen, the application simply starts as a host and does not broadcast its IP[8] address over computer network nor actively searches for existing sessions. Unity Cloud option utilizes the official Unity Multiplayer service to provide the possibility to interact via the internet. In this case, when starting a server, match with the preset name is created on the Unity servers, whereas when trying to connect as a client, the application tries to find internet match with the same name and connect to it. The advantages of Unity Cloud are mainly possibility to connect two computers on different local networks and also the fact that it offers a free plan. The disadvantages are that this service is being deprecated (together with whole UNet technology) [87] and that some of the limitations, like a bandwidth limit, of the free plan are quite challenging. When starting as a host on LAN, broadcasts are sent in regular intervals to inform other computers on

8. IP – Internet Protocol

the same network about the running network session. Broadcasted data include the application name to distinguish between different applications broadcasting with the same settings. When trying to connect as a client on LAN, the program listens for broadcasts, and when compatible one is received, it joins the session running at the received IP address of the server. VPN option utilizes ZeroTier technology [88] and requires Zero Tier One client to be installed on all computers. On the website of the ZeroTier, a virtual network is created with a unique identification. This network is free (with some limitations regarding the number of users) and is available nonstop, considering ZeroTier servers are running. For the computer, it behaves like a local network although it spans across the internet. All connected computers have therefore their unique IP address (from range which is defined on the ZeroTier website in the network settings) and can see and ping each other considering no firewall or other mechanism is preventing this communication. The users of the application do not have to bother about this network at all. When the user selects a VPN option, the application (in the background) calls Zero Tier One client via a command line and passes arguments to join the network. When the option is deselected, or the application is exiting, the network is left using the same approach. The application implements some mechanisms, like check for command line exit codes and search through all IP addresses computer possesses, to determine whether the state of the ZeroTier connection is valid. When the computer successfully joins the ZeroTier network, everything then works the same way as in the case of LAN above. The only difference is that different IP address and port number is used for broadcasts and when connecting as a client.

### 4.4.3 User Avatars

The underlying CVE platform provides user representation. Each user is represented as a simple avatar (as shown in Figure 4.2) with the HMD and VR controller in hands. Both the positions of the avatar and controller are synchronized over the network. Although the avatars are very simple, they allow for some core interactions between users like waving with a controller to express greetings, and also shaking or nodding of the head to show disapproval or approval.

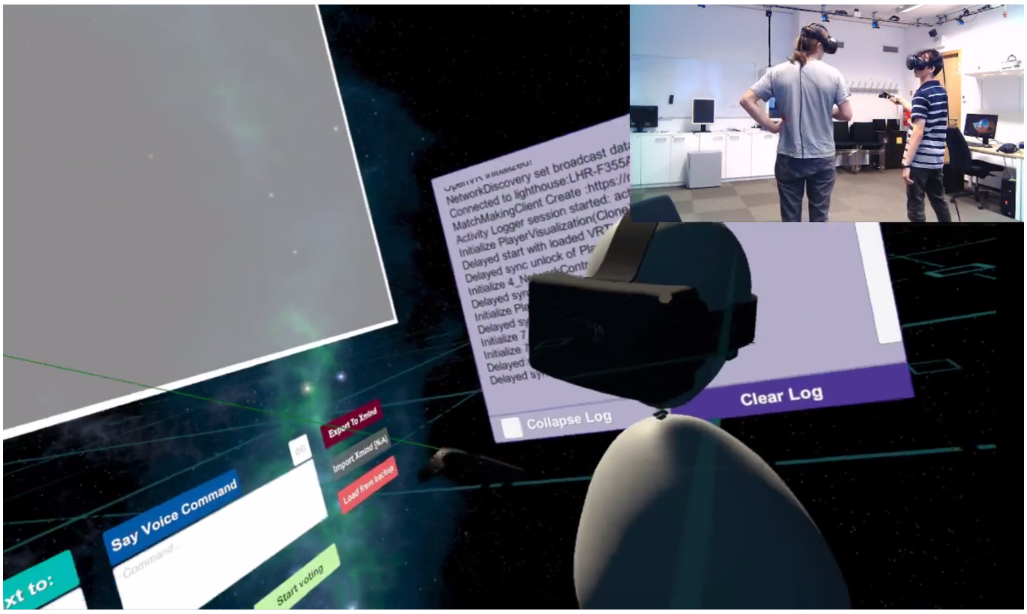


Figure 4.2: Representation of the user in VR environment with overlaid image of real users

## 4.5 Environment

During the development of the application, two different environments, positioned in space, respectively office, were used as shown in Figure 4.3.

The space environment was present in the application from the beginning and was also used in the tested version of the system. It placed users inside the endless space providing simpler visuals aimed mostly at the system functionality and feeling of "freedom without boundaries". However, when one user of greater age tried the system (informally, i.e., not as a part of user study described in chapter 5), he tried to look down which caused him slight nausea. Therefore it was decided that adding a floor might be a good idea to prevent potentially similar problems in the future. Moreover, since it would not be very visually pleasing to have a space environment with just a simple "floating" floor, the whole environment was redesigned as an office space. Freely available 3D models from the Unity Asset Store

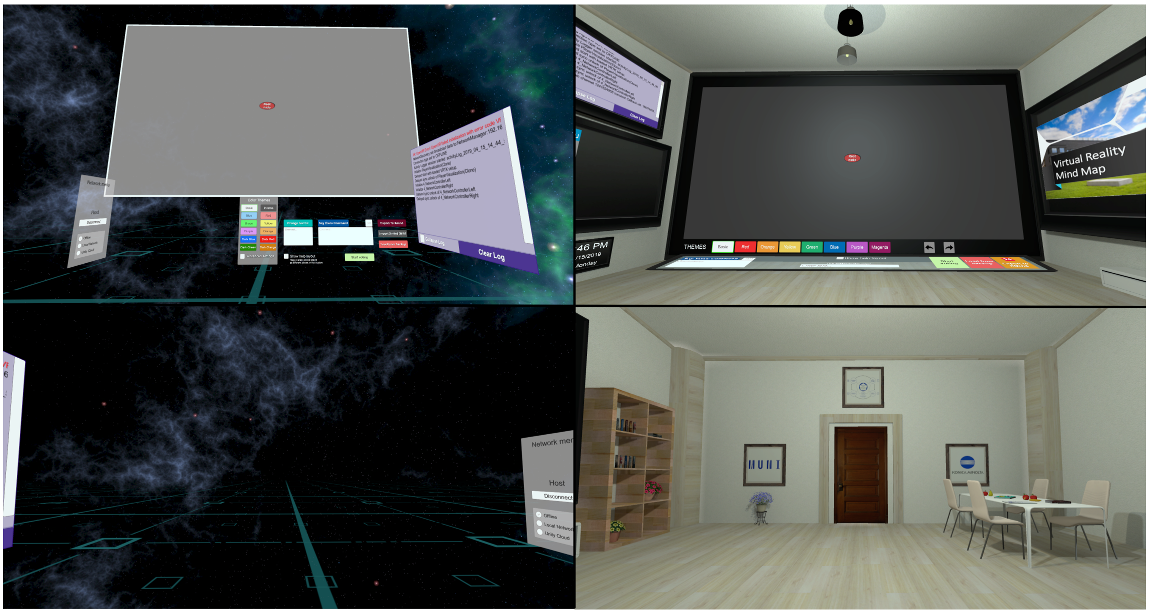


Figure 4.3: Initial and current VR environment

and related sources were used to compose the new visuals. Designing a new realistic environment posed some new challenges. The first one was related to the fact that environment scaling issues are much more noticeable with the VR HMDs than in the case of traditional monitors. [89] It was, therefore, necessary to get the size of individual elements right. As it turned out, mind map canvas in the space version was huge and prototype of the room with adequately large walls shown that this type of space does not feel very comfortable as it is too overwhelming. Consequently, dimensions of mind map canvas and nodes were reduced. At the same time, the rendering quality of the mind map was improved to make sure that the readability is decreased as little as possible. Another challenge of new environment was necessity to rearrange user interface to fit into the environment while still being well visible. Finally, better lighting was added – the office environment utilizes two baked lights, with a yellow, respectively blue, tint, to achieve the desired feeling.

## 4.6 Mind Map

As mentioned in the section 2.1, a mind map can be described as a set of nodes and relations between them. This description also determines the core elements which need to be somehow represented in the underlying codebase. Figure 4.4 shows a screenshot from the application containing an example mind map with several nodes and their relations.

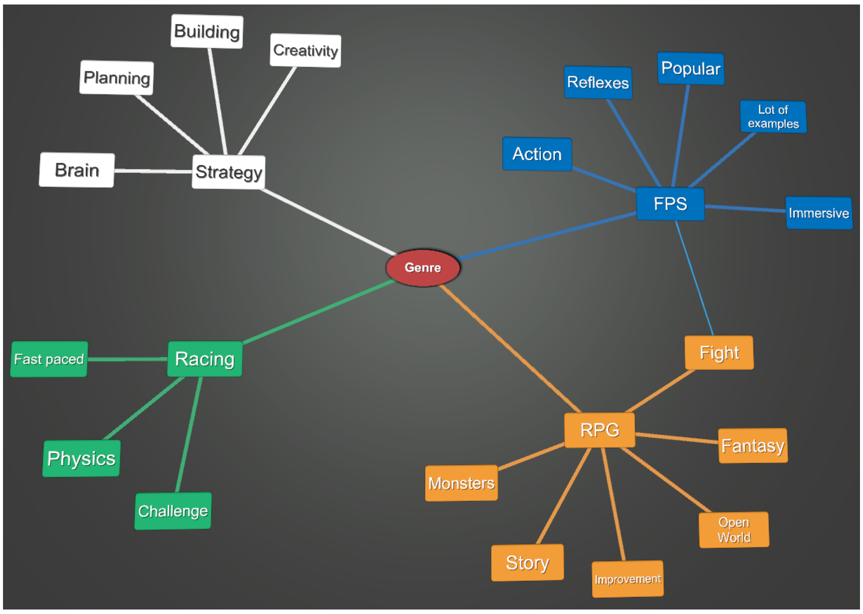


Figure 4.4: Example mind map created in the application

### 4.6.1 Mind Map Node

Map node is the cornerstone of a mind map. Each node is represented by a visual element having a specific color and a text label. Node properties can be modified using a combination of user interface below the mind map canvas and radial menu which opens when a user points at the given node. Nodes can be freely moved around with a drag and drop mechanism. As can be seen in Figure 4.3, the user interface below the canvas contains buttons related to color themes. When user presses one of these buttons, all selected nodes will be recolored. Figure 4.5 (number 1) shows the appearance of the node's radial menu. Blue buttons provide functionality to add relations to other nodes. Red button removes the node while green button creates a new node as a child of currently selected node. The last button allows users to record a text for this node.

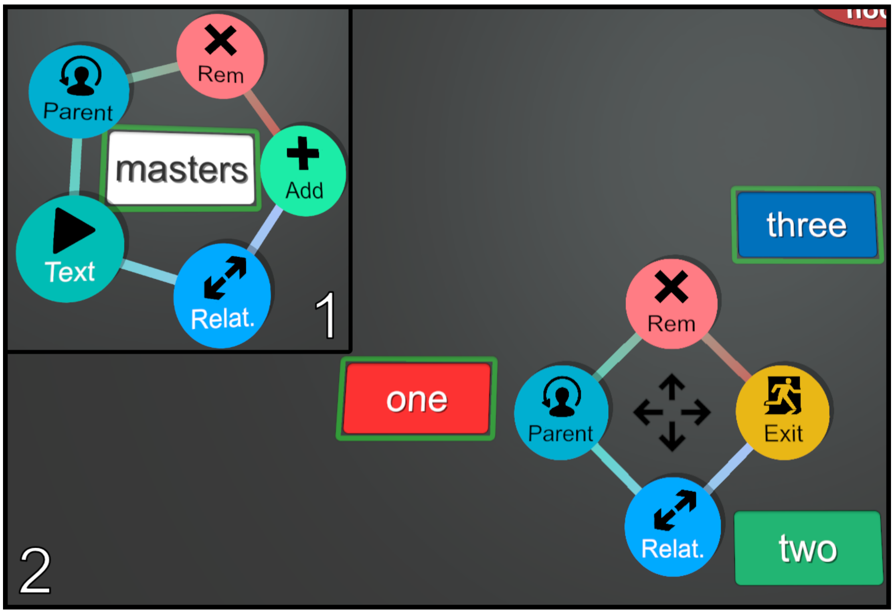


Figure 4.5: Radial menus for a single selected node (1) and multiple selection (2)

At the moment, there are three ways how to create a node; this feature therefore also exhibits multimodal characteristics:

- Use the button in the radial menu.
- Point at some blank space in the mind map canvas and wait for a while.
- Draw a closed shape without a node inside (explanation of the "drawing" technique is in section 4.6.2).

The reason why there is more than one way is that it allows the user to choose the most suitable approach. Regarding the relations, there are two types implemented. Both of them are binary, i.e., between two nodes, and are represented as a line.

- Parent/child relation
  - Each node can have only one parent and as many children as desired. No two nodes can therefore share the same child.
  - The node is dependent on its antecedents – when some of them are moved or removed, this node is modified as well.
  - Visualized as thick line with the same color as the child
- General relation
  - Each node can have as many of these relations as desired. More nodes can be therefore related to the same node.
  - Nodes in this type of relation are independent. When one of them is moved or removed, the other one is not affected.
  - Visualized as thin line with blue color

Thanks to these two slightly distinct types of connections, it is possible to describe the relations between nodes with more precision.

### 4.6.2 Multiple Selection

Multiple selection allows the user to select more nodes at once in order to modify them. It is implemented in such a way that the user is required to draw a shape (as visualized in Figure 4.6) around the nodes he or she wishes to select. Selection shape is drawn with a controller, starting with switching the controller's state to "selecting" (as described in section 4.3) when pointing at a blank space in the mind map canvas. After that, the user can draw the shape just by moving the controller while pointing at the canvas. When the selection is finished, it is evaluated. If there are no nodes inside the shape, and both of its ends are close to each other, a new node is created at the endpoint's position. If there are some nodes inside the shape, they are selected, and the user can move them or execute some other actions provided by the appropriate radial menu shown in Figure 4.5 (number 2). Multiple selection, therefore, simplifies the modification of properties of several nodes at once. In the background, this feature is based on the point-in-polygon test. The selection shape is visually represented as a long red line, or more precisely, polyline, which is converted into a polygon (based on vertices of individual line segments of a polyline) after the drawing is finished. Then, for each node, it is computed whether its center lies inside the resulting polygon or not.

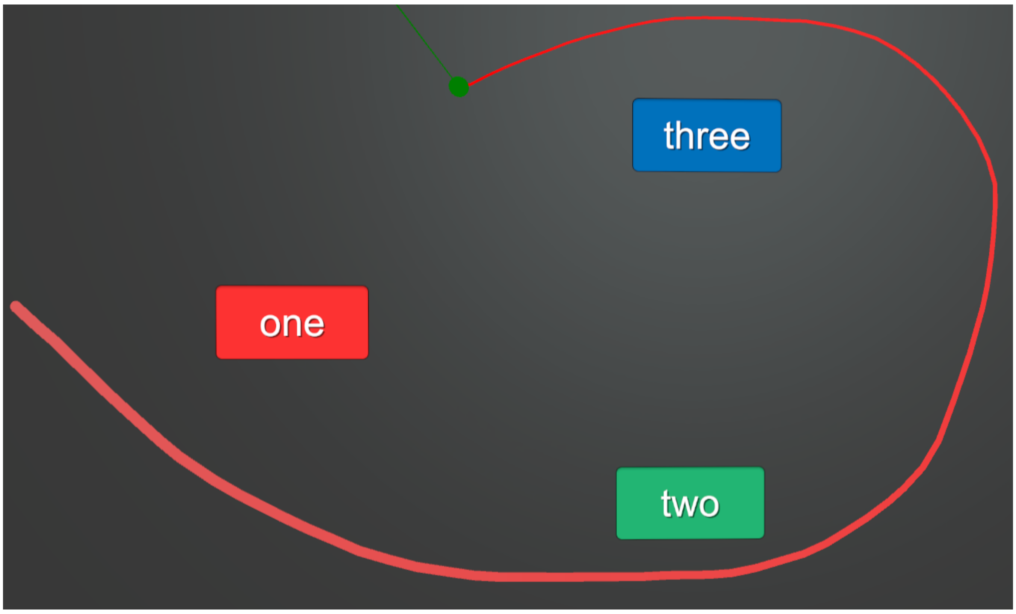


Figure 4.6: Zoomed-in image of multiple selection in progress

### 4.6.3 Voting

Since the purpose of this application is not to provide the ultimate mind mapping solution, but mainly to use mind maps as a brainstorming tool, it offers features supporting this goal. One of them is voting, which can be executed as the last step of the brainstorming, allowing the participants to determine the best ideas. The voting feature recognizes two types of users – moderator of the brainstorming, who is the person hosting the networking session and leading the voting, and participants, i.e., the rest of the users who are actually voting. The moderator starts the voting by using the appropriate button. The user interface of the application then slightly changes allowing the moderator to choose colors to vote about and the number of points each participant has to distribute. The voting consists of voting rounds; in each round, one color is voted about. During the voting round, participants see only the number of points they assigned, and they have to distribute all points. A text box and plus and minus buttons are attached to each node to simplify this process as can be seen in Figure 4.7. When the moderator tries to end the voting round, the system allows it only if all participants distributed assigned points. When it seems no more voting rounds are needed, the voting can end. In this case, all users see the summary of points for each node while winners in each category, respectively color, are made visually distinct.

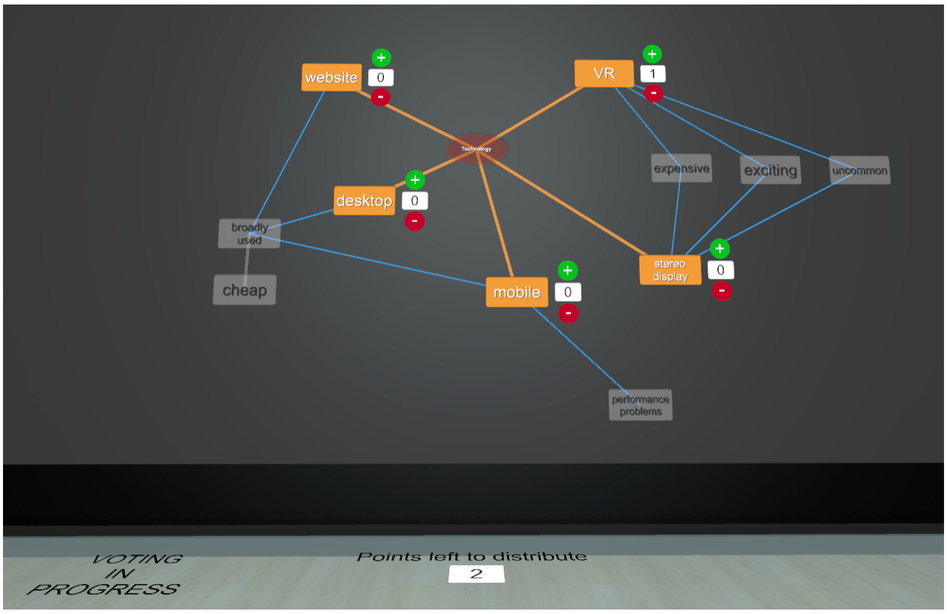


Figure 4.7: Participant's interface during the voting round regarding the orange category

### 4.6.4 Undo/Redo Operations

Undo/redo functionality allows users to revert some of the changes made to the mind map. This feature was implemented in the later versions of the system following the outputs of the user study (chapter 5), which revealed that it would be good to have some of the actions, like a change of a node's text, reversible. Regarding the implementation, it is based on the command pattern with some extensions to support networked operations. Since there are many ways how to implement such a feature, it is interesting also from the system design's point of view. Simplified UML class diagram is shown in Figure 4.8. Each undoable operation is encapsulated as a command, i.e., class deriving from IAppStateCommand interface providing functions with self-explanatory names – Execute and Unexecute. Commands are designed to run only on the server, and it holds that if the command is executed or reverted, changes are propagated to all clients. All commands are managed by AppStateUndoRedoManager class providing the possibility to undo, respectively redo, actions thanks to two stack data structures holding information about the next command to execute.

Regarding the network implementation, the host's MindMapManager holds separate AppStateUndoRedoManager instance for each client; this means that each client has its list of commands which are stored on the server. When a client wants to perform undo or redo, it sends a request to the server, which calls the desired method on the corresponding AppStateUndoRedoManager instance. Thanks to this, each client reverts, respectively executes again, only his or her actions. Commands also contain GetInfluencedNodes method returning the list of nodes being modified by the given instance of the command. This functionality enables to remove some commands from the stack when they are no more valid, e.g., when all nodes they influence were already removed.

### 4.6.5 Export And Backup

Because the application provides similar mind map visuals as standard software tools, it was clear from the beginning of the development that it would be useful to have the possibility to export the results for later examination. There are several formats designed especially for mind maps, XMind File Format[9] was chosen as the one to be implemented in the end thanks to the good documentation and XMind application being highly popular. This format is basically a ZIP archive with several XML files describing the mind map content and visuals. Figure 4.9 shows juxtaposed mind map in the application and the exported version previewed in the XMind. All of the most important information are exported including the number of points the node received during the voting. What is not saved, however, are the positions of nodes which have a parent and the parent is not a root node. For these nodes, XMind forces its own layout so it is not possible to position them freely.

To prevent loss of all data in case of unexpected application exit or related problems backup is implemented. The backup file is automatically created in regular intervals, and the application contains a button allowing to load the last one. Custom format for storing backup data was designed – it is a text file where each line contains a serialized JSON[10] data of one of the nodes. There are two main reasons why is such file format used – it is quite lightweight and thus has a good performance while being also well manageable from the perspective of implementation.

9. XMind File Format – more at `https://github.com/xmindltd/xmind/wiki/XMindFileFormat`
10. JSON – JavaScript Object Notation

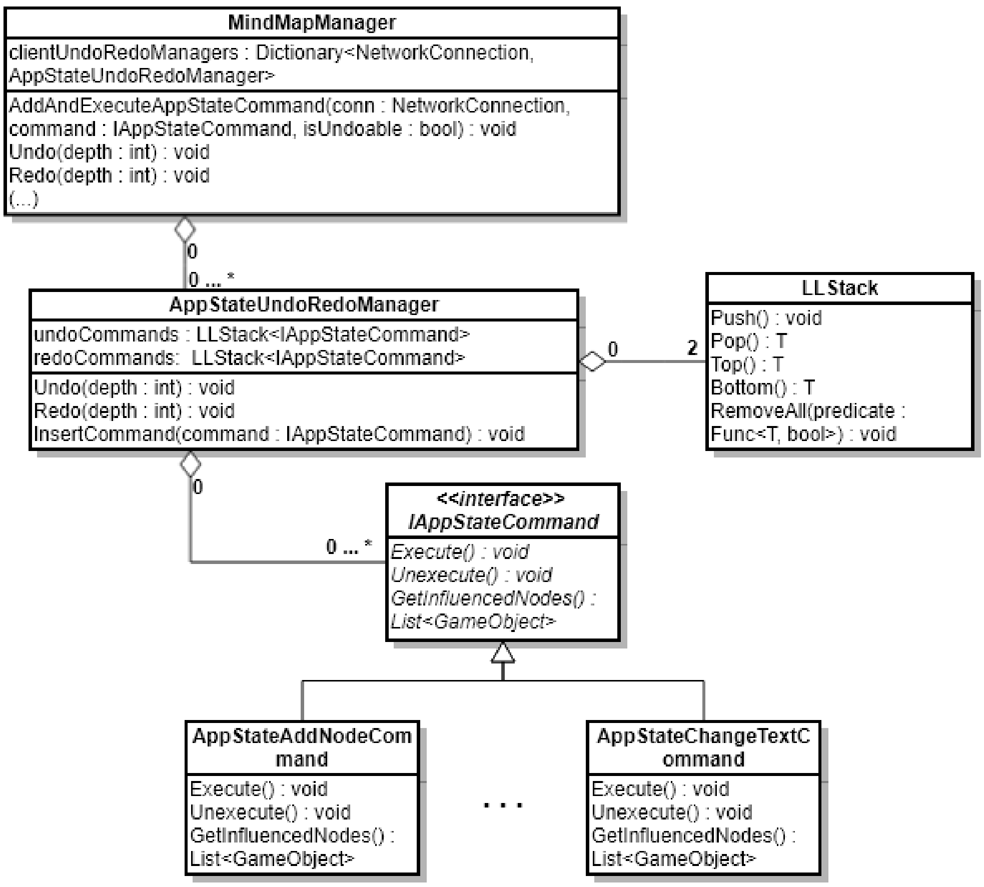


Figure 4.8: Simplified UML class diagram of the Undo/Redo functionality

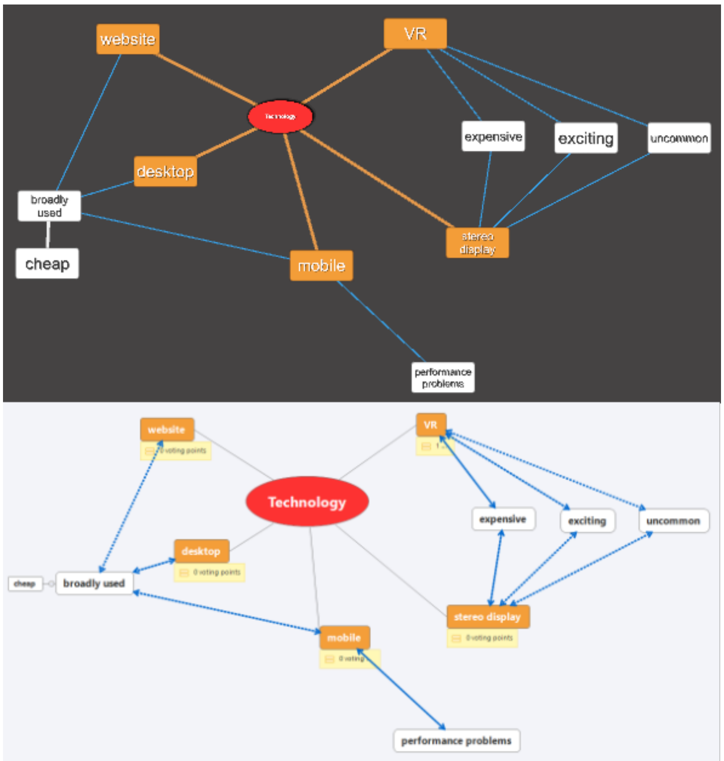


Figure 4.9: Mind map appearance in the application and when exported and previewed in the XMind

## 4.7 Voice Chat

Since the beginning of the development of the application, Skype was used for voice communication. However, in the latest version of the system, Skype was replaced with an integrated solution, based on service Agora.io, [90] so no third party software is therefore needed for this purpose anymore. The implementation is using Agora's Unity SDK providing all of the core functionality. When the application wants to start a voice chat, it needs to join a channel identified by a number. Currently, it works in such a way that the host of the networking session generates a channel identification and joins this channel. When some client joins the game, the voice chat codebase waits until it receives the identification from the server. As soon as that happens, it joins the corresponding channel, and the user can talk with others. Using integrated voice solution over third-party application is a huge advantage for future development because it gives the application power over the audio. For example, thanks to this, audio can be muted if desired. Regarding the price, Agora.io offers 10,000 minutes for free each month, which is more than enough for this application.

## 4.8 PDF Viewer

Since the application provides a whole virtual environment, there exists a space for additional interactive elements. From the perspective of brainstorming, it would be useful to have an ability to view presentations which may help to guide the ongoing session or provide a summary of the discussed topic. For this reason, support for previewing PDF files was added during the development of the office environment version of the system. This file format was chosen for its widespread usage as it is a unified container for documents and presentations across many software tools. The screenshot of the office environment in Figure 4.3 shows the PDF Viewer on the right wall of the room, i.e., next to the mind map canvas.

Unity does not offer the possibility to load PDF files by default, so it was necessary to come up with a custom solution. After the research of libraries supporting this file format, MuPDF was chosen as the one to be used thanks to the AGPL[11] license and indisputable qualities achieved by a long development and vast community of users. This library is written in C, so the first necessary step was to create DLL files and write appropriate C# wrappers to make it possible to access the library functions from the C# code. The next step was to decide how to process the PDF so it will be possible to show it in Unity. Converting the PDF into several PNG[12] files, where each one corresponds to a single page of the source PDF, turned out to be the best option. Therefore, the purpose of the MuPDF library in the whole process is to take the input PDF and convert it into a set of images. As soon as this is done, these images are loaded as textures into the Unity. The environment contains a predefined position where the result is shown together with buttons allowing to list through pages.

Regarding the decision what PDF to load, when the application starts, it looks for a file with a name "presentation.pdf" located in the same directory as the executable. If this file exists, it is automatically loaded. Otherwise, nothing happens. To simplify the distribution of the file to the rest of the brainstorming session participants, the network host automatically uploads the file to the server via FTP[13] and sends the name of the file on the server to clients. Afterward, clients automatically download and process the PDF. Therefore, they do not have to own the file before joining the network session. The file is uploaded to the external server via FTP, instead of sending it directly to the users via a Unity Networking layer, because of bandwidth limits of the Unity Cloud service.

11. AGPL – GNU Affero General Public License
12. PNG – Portable Network Graphics
13. FTP – File Transfer Protocol

# 5 Testing

This chapter presents the results of the performed user study. As mentioned in previous parts of the text, the space environment version of the system was used for the testing, some of the features added later on were therefore missing. The user study was approved under reference number EKV-2018-093 by the Research Ethics Committee of Masaryk University.

## 5.1 Purpose

The primary purpose of the user study was to verify the usability of the system, i.e., find its strengths and weaknesses. Although some occasional informal testing was performed already during the development of the application, it was hard to predict how will the system perform in the hands of other people. Will they be able to control it? How usable will the speech to text service be? What about the quality of collaboration? Will they have some problems regarding readability? Is someone going to feel nausea or other related sicknesses? These are some of the questions asked before the study. The expectation was that participants of the study would be able to create a simple mind map while they will probably have to overcome some challenging situations related to speech to text technology.

## 5.2 Methodology

### 5.2.1 Participants and Questionnaires

The study consisted of a total of 32 healthy participants (19 male, 13 female), all of them were native Slavic speakers. Since the testing was happening in pairs, this makes for 16 testing groups. Participants were a voluntary sample, recruited based on their motivation to participate in the study. All of them were between 18 and 33 years old and regular computer users. They were rather inexperienced with mind maps and generally had some experience with remote collaboration.

Regarding the process of the study, the very first step was to explain the workflow of the experiment to participants. Afterward, subjects filled in the following documents:

- **Consent form** signed by all participants confirming their consent to participate in the study and to publish their anonymous data.
- **Form with demographic data** containing questions involving user's age group or computer skills.

After this step, the testing took place as described in section 5.2.2. As soon as the experiment was finished, subjects were asked to answer questions related to the recent experience. Following questionnaires and types of questions were used:

- **Presence questionnaire** [91] [92] consisting of nine quantitative questions answered on a 7-point Likert scale.
- Three **questions about knowledge of similar systems** and their advantages, respectively disadvantages, when compared to this application.
- Set of eight questions related to **evaluation of system features** using a 5-point scale.
- **NASA Task Load Index** [93] (hereafter TLX), measuring cognitive workload, consisting of six questions answered on a 21-point scale.

The subjects were also asked to fill in a free-form debriefing session questionnaire, where they provided qualitative feedback for the whole experiment. All questionnaires used during the user study are part of this thesis; more information about their location is provided in Appendix A.

### 5.2.2 Procedure

The process of user testing consisted of two main steps. Participants were located in different rooms, and during the first ten to fifteen minutes, depending on the skill of the individual user, each of them was alone in the virtual environment while being introduced to the system and presented with its features. His or her feedback was gathered while trying the system functionality. The second part of the evaluation consisted of participants trying to brainstorm on a scenario. As the first step, Skype call was initiated with both of the participants and instructor, i.e., me. Participants then joined a common brainstorming session inside the application. To assess the functionality of the system several different brainstorming scenarios were designed:

- How to cook an egg properly.
- What is best to do on Friday night.
- How will artificial intelligence take over the world.
- Wine selection for dinner.

There was also an idea that participants may choose whatever topic they want. However, it was decided that to get a comparable and more consistent data, only one of these topics should be chosen and used for the study. After thorough consideration, the second one, i.e., "What is best to do on Friday night", was chosen since it seemed to be the most general and easygoing scenario. The whole process of brainstorming was directed by the instructor and consisted of the following steps:

1. Participants were asked to present possibilities on how to spend Friday night by placing appropriate nodes on the mind map canvas while collaborating.
2. Participants were asked to assign other specific properties to ideas from the previous exercise and to use a different color of nodes.
3. Each participant was asked to select one idea and add nodes describing a concrete proposal.
4. Participants were asked to present to each other results of the previous exercise.
5. Participants ran a voting session with two rounds. One of the participants took the role of a voting moderator; the second one was acting as a participant.

Time of completion for each of the steps was measured, and the behavior of the participants was monitored in order to get another type of feedback.

## 5.3 Results

This section presents the outputs of the user study. First, qualitative results are discussed, i.e., feedback obtained from participants is presented. What follows is the presentation of quantitative results summarizing point-based questions and relations between answers. Finally, the most interesting outputs of logs are presented revealing some of the users' behavior.

### 5.3.1 Qualitative Results

Participants provided valuable feedback important for further improvements. The feedback was gathered not only by direct communication with participants and via questionnaires but also by watching their behavior during the ongoing scenario. Thanks to this approach, it was possible to collect useful information during the whole time of the testing.

The first type of feedback to be examined will be responses to question about knowledge of similar systems for remote brainstorming and collaboration. Subjects were also asked to describe their main advantages and disadvantages when compared to this application. Users mentioned a variety of different tools, some of them are as follows: Skype, Google Docs, Facebook Messenger, IBM Sametime, Slack, Discord, Cisco Webex, Microsoft Outlook, and GitHub (this one seems a bit misjudged at first sight, but it is true that in the end, it is a good example of a remote collaboration platform).

The most commonly mentioned advantage of our system was immersion. Quoting one of the users, "It makes you feel like you are brainstorming in the same room on a whiteboard (...)". Related to this is "no outside distractions" which appeared several times as well. Participants also appreciated that everything is happening in realtime, so the application provides instant visual feedback. User avatars were also mentioned as an advantage and better solution than "just rectangles with names" as is common in some applications. One user also stressed out the utilization of mind maps, which is something applications he or she knows do not offer. Several subjects also remarked that this application is more intuitive and more personal, quoting one of the participants: "As close to the real (in person) meeting as possible." What was also mentioned many times is that this experience is "more fun". This paragraph will be concluded by quoting one participant who wrote that "It's definitely more involving, with good knowledge of the tool you are able to get better results".

As for disadvantages, many users pointed out the necessity to have appropriate hardware, i.e., that such an application requires more equipment and preparations than the tools they know. VR HMD was also considered by some users as "not so comfortable" and causing physical discomfort, possibly also aches after longer usage. Some users would wish to see the participants' faces. Speech to text was also mentioned by some subjects since it is less effective than keyboard typing. The application also has a steeper learning curve according to some of the users.

Participants were also given the possibility to provide us with additional comments related to the experiment and any part of the system. Regarding the comments related to the experiment itself, they were mostly positive, for example, "Experiment was well organized", "When can we test again?" and "It was a cool and enjoyable experience!". One user wrote that "It would be good to have some testing topic which requires more discussion between participants, something more challenging/real". The application received positive feedback as well; for example "app is super-responsive" and "good work with color". The most mentioned drawback of the system was the position of the mind map canvas. It was located too high forcing users to look up all the time causing physical discomfort as well as decreased readability of nodes which were positioned at the top. This problem was even more noticeable for one of the participants who was sitting on the chair all the time instead of standing. Another thing to improve, noticed by observing users' interactions with the system, was that the possibility to revert certain actions might come as handy. Users generally spoke favorably about the space environment as it made them feel relaxed and was not distracting. Some subjects had complaints about the speed and reliability of the used speech to text service. The user interface has space for improvement as well. Mainly at the beginning, users tended to forget to stop the voice recording after they finished saying the desired text for a node. Also, the difference between parent-child relations and general relations was not clear enough. Some feedback was involving the controller and possible improvements in controls – for one user, it would be more suitable to use the toggle technique for the laser pointer and thus do not require pressing the trackpad button all the time. Users also provided some ideas for further improvements. One participant mentioned that it would be good to have the possibility to decide whether to hide or show activity (including laser pointers) of other people during the voting. Other one added that the voting session would benefit from a list of participants who did not distribute their points yet, so the moderator instantly sees the current progress. One tester said that it might be useful to have more control over voice so the user can mute himself/herself or others, for example when saying a text for a node. This need was also observed from the participants' interaction during the study – although the system does not have problems when more participants use the speech to text at the same time, the fact that they hear each other talking on Skype is confusing and making them feel like they might be messing up speech recognition for the other user. Ability to change the size of the node's text would also be welcomed addition for some users.

Generally said, based on this feedback, the application seems to be quite immersive but for the price of increased physical demand and possibly slower pacing due to voice recognition.

### 5.3.2 Quantitative Results

This part of the text looks at the data obtained from questionnaires from the statistical point of view and reveals some of the most valuable findings. First discussed topic will be the performance of participants.

The median of collaboration times for the scenario was 19 minutes and 5 seconds (excluding explanations of each step). Since nodes were created only during the first three steps of the scenario, the median of these times is important as well; it is 14 minutes and 20 seconds. This accounts for an average speed of approximately one to two nodes per minute as the median of a number of nodes created during the scenario was 24. It is worth mentioning that the speed, respectively duration, of

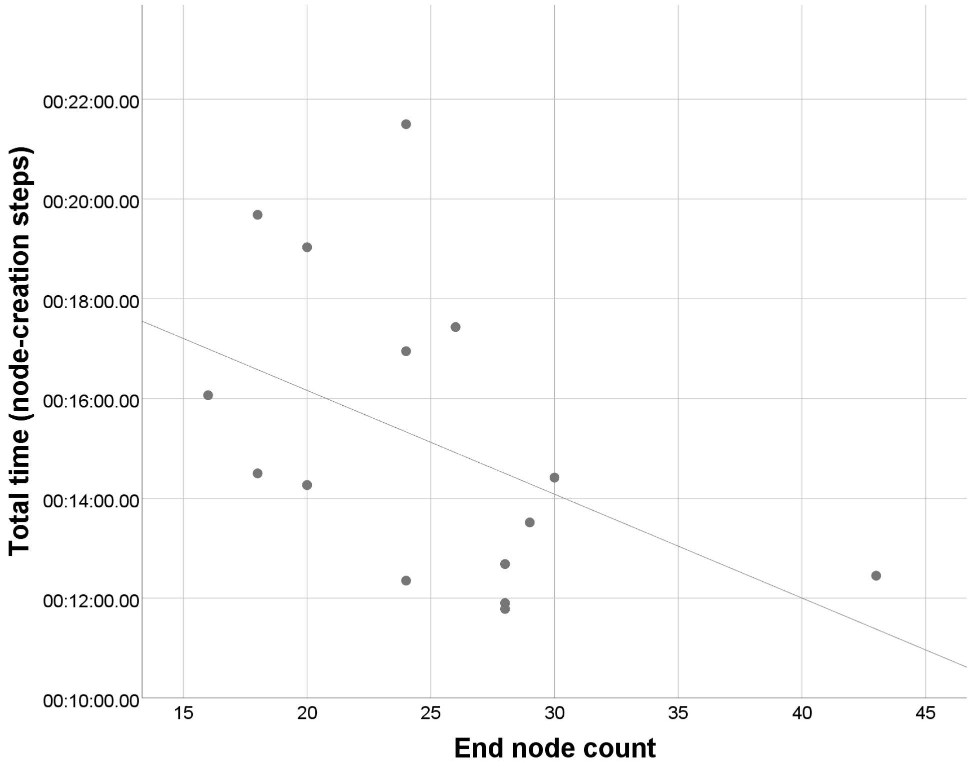


Figure 5.1: Scatter plot of collaboration times and number of nodes

the brainstorming depends on the creativity of the users. The fastest pair was able to create 3.3 nodes per minute on average while the slowest one achieved the speed of almost one node per minute. When the relation between the number of nodes at the end of the exercise and total time is graphed, the result is quite interesting as is shown in Figure 5.1. The observed behavior can be explained in several ways. First, the users with higher node count might have been simply more creative than the rest, and so it was easier for them to come up with new ideas. Moreover, as each step of the study was not limited by time but rather by a rough minimum of the number of nodes, these participants had no problems creating more than enough nodes to continue, and the flow of the session was not interrupted so much by the time spent on thinking about possible ideas. The results can also be caused by the possibility that slower participants might have chosen more complicated topics and therefore had more problems with speech recognition. However, no significant correlations were found in this regard. The effect can also be caused by differences in the effectivity of communication between participants. In any case, this confirms that the speed of the brainstorming does not depend only on the system capabilities.

Users were also asked to evaluate the usability of system features; the most important outputs will be now examined. Figures 5.2 and 5.3 confirm observed behaviour which is that users had no major problems when trying to create or delete nodes. The delete might perform a bit worse because when a node is deleted its radial menu remains open until the user points elsewhere. Although the menu is not working anymore, it is a bit confusing that it is still present. This behavior is going to be fixed so the application can deliver a smoother user experience. Distribution of responses in Figure 5.4 shows that the mechanism for node movement was not as user friendly as desired for some participants. This might be caused by the fact that moving of a node fails, i.e., the node returns to its starting position, when the trackpad button is released before or at the same time as the trigger. This was a slight complication for some users. This is one of the reasons why the latest version of the system implements updated controls. Values in Figure 5.5, containing evaluations of "color change" feature, cover the whole five-point scale. The results, in this case, are also rather positive but not as good as was expected. The explanation could be hidden in the way this feature works. When a user selects a node and wants to change its color, he or she has to press the corresponding button in the user interface below the mind map canvas. The first observed disadvantage of this approach is that this feature is not present directly next to the node, but the user has to look somewhere else. Moreover, since the mind map canvas is huge, the color themes interface might not be visible at first sight, so it is necessary to realize where to find it. The second thing, which happened several times, is that when the user moves with the laser pointer from the node to the color button, some other node on the way can be mistakenly selected during this process and therefore recolored instead of the intended one. From the user's perspective, this can be easily avoided by turning off the laser pointer during the transition. However, since some users were not keen on doing this, it might be necessary to think about design improvements in this regard.

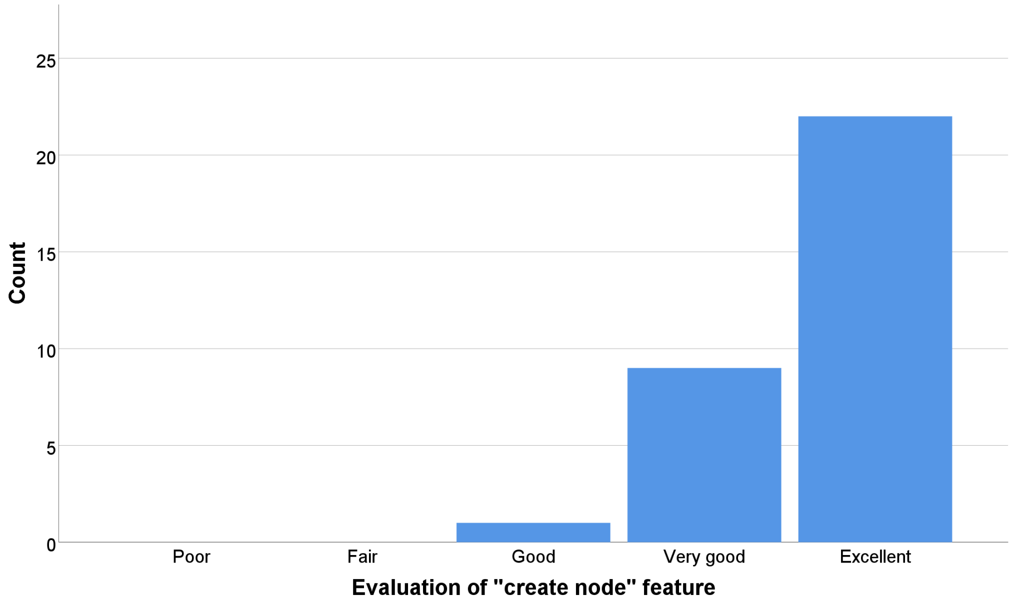


Figure 5.2: Evaluation of usability of node creation feature

Figure 5.6 shows the histogram of answers to question about satisfaction with speech to text technology. Histogram of evaluations of general "change text" functionality follows a similar distribution. The collected data support statements from qualitative feedback (section 5.3.1). The mean of the distribution is 3.94 while the median is 4, it can be therefore said that the speech recognition in its current state is acceptable. The question is, how would it perform if different scenario with more complicated words was used? Hence, although the performance is not bad, there is a space for improvement. Starting with a user interface which can be adjusted to avoid problems mentioned in section 5.3.1. Then, it might be worth considering whether to stick to the current speech recognition solution or try something else. Another idea to think about is to utilize multimodality even in text input. It was not unusual that the user said a word which was recognized as a different one, but also very similar, to what he wanted so the difference was just a few letters. It might come as handy to have a quick way of

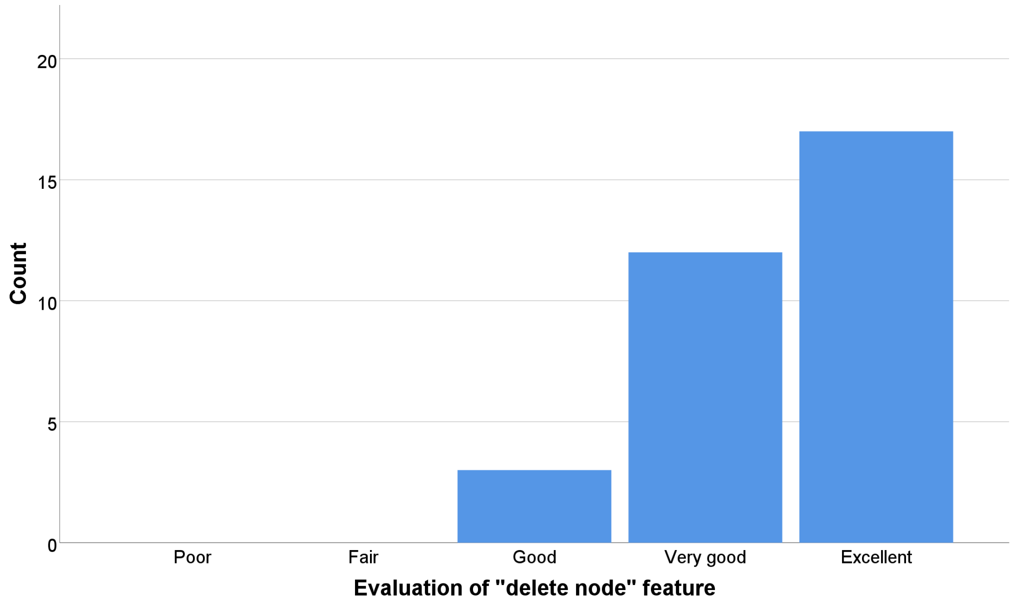


Figure 5.3: Evaluation of usability of node deletion feature

fixing these errors, either in the form of a virtual keyboard or some dictionary-like mechanism.

The second part of this section presents the most interesting outputs obtained by computing Spearman's correlation above the data. Tables 5.1 and 5.2 present significant correlations between questions about ease of the VR cooperation, adjustment speed, and willingness to use the application for the next meeting. Correlation between the last two questions is not in the tables, but is significant as well – coefficient being 0.540 while significance is 0.001. These data can be interpreted in the following way: if the users are able to adjust to the environment quickly, it means that the following cooperation will be easier since they feel comfortable and in control of the application. Everything seems more natural to them. Moreover, when the cooperation is smooth and feels easy, users are more willing to use the application for their next meeting. All of these correlations show the importance of making the application as accessible as possible so any user can quickly use it without going through complex tutorials. User's experience gained inside the application is critical in deciding

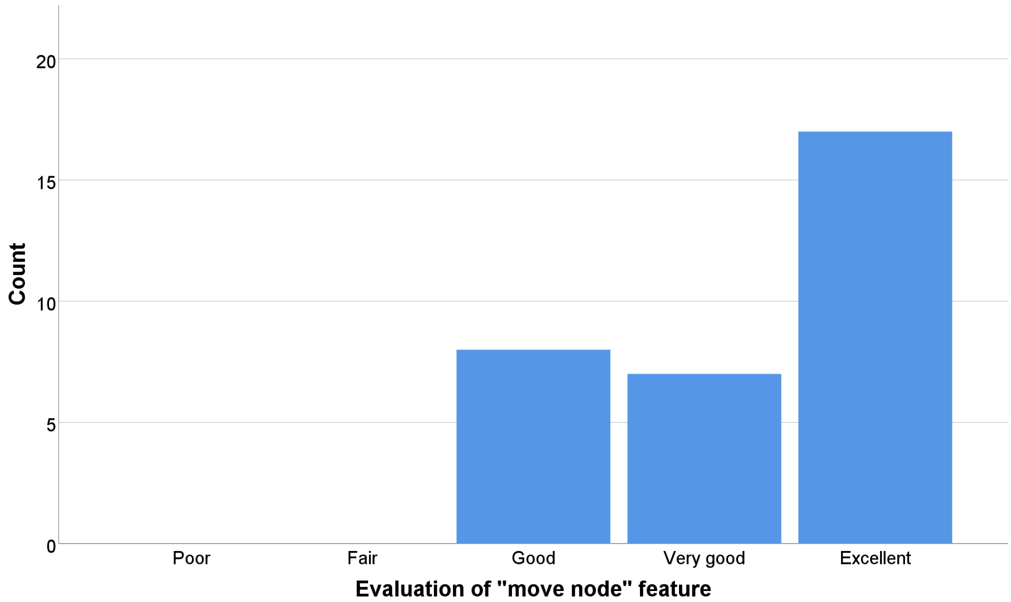


Figure 5.4: Evaluation of usability of node movement feature

whether he or she is giving this application another chance next time. According to the data, this decision is in this case also influenced by the quality of the "text change" feature – there is a correlation of 0.353 with significance 0.048. This finding shows that the final judgment is based on an overall feeling involving the cooperation, environment, and learning curve while emphasizing the quality of the "text change" functionality which influences the real usability of the application a lot.

Another interesting correlations, related to physical demand and frustration, are presented in Tables 5.3, 5.4, 5.5, and 5.6. Table 5.3 shows that there is a positive correlation between the physical demand and frustration. When the user felt physical discomfort, caused for example by too highly placed canvas or weight of the HMD, he or she became more frustrated. Since there also exists a negative significant correlation (coeff. -0.384, sig. 0.03) between the frustration and willingness to use the application for the next meeting, it is clear that the probability that the user will feel frustrated should be minimized. Physical demand can be partially decreased by improving the application's

| | |
|---|---|
| **Question SC1:** | How easy did you find the cooperation within the environment? *scale: worst-to-best* |
| **Question SC6:** | How quickly did you adjust to the VR environment? *scale: worst-to-best* |
| **Correlation:** | 0.528 |
| **Sig. (2-tailed):** | 0.002 |

Table 5.1: Significant correlation between questions about ease of VR cooperation and adjustment speed

| | |
|---|---|
| **Question SC1:** | How easy did you find the cooperation within the environment? *scale: worst-to-best* |
| **Question SC9:** | Would you use the application for your next virtual/remote meeting? *scale: worst-to-best* |
| **Correlation:** | 0.470 |
| **Sig. (2-tailed):** | 0.007 |

Table 5.2: Significant correlation between questions about ease of VR cooperation and use for next meeting

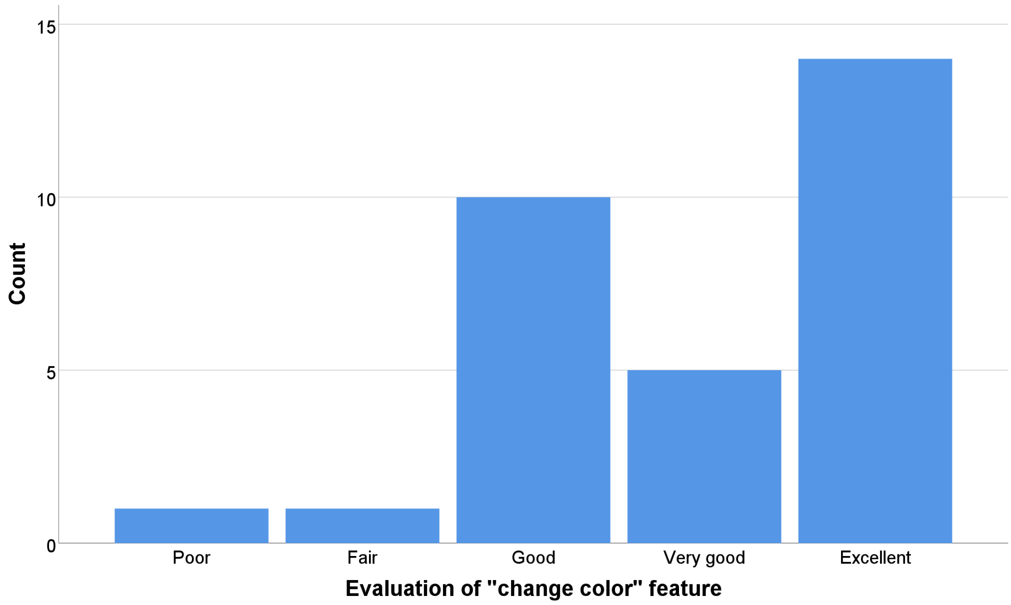


Figure 5.5: Evaluation of usability of "color change" feature

interface, but as long as HMD is used, there will always be a certain level of discomfort. Not just from the perspective of wearing a helmet on the head, but also from vision's point of view since, as is shown in Tables 5.4 and 5.5, the insufficient visual display quality of HMD correlates with physical demand and feeling of frustration. This is a confirmation of technological limits of current HMDs. The best solution at the moment is to improve the colors and sizes of user interface elements to decrease the users' eye strain caused by the relatively low pixel density of the headset. Last interesting correlation regarding the frustration, shown in Table 5.6, is involving the "change text" functionality. Since this is very important feature from the system perspective, it is used a lot, and when the user does not feel comfortable with it, it might make him or her frustrated.

| | |
|---|---|
| **Question TLX2:** | How physically demanding was the task? *scale: best-to-worst* |
| **Question TLX6:** | How insecure, discouraged, irritated, stressed, and annoyed were you? *scale: best-to-worst* |
| **Correlation:** | 0.479 |
| **Sig. (2-tailed):** | 0.006 |

Table 5.3: Significant correlation between questions about physical demand and frustration

| | |
|---|---|
| **Question SC8:** | How much did the visual display quality (HMD) interfere or distract you from performing the assigned tasks? *scale: best-to-worst* |
| **Question TLX2:** | How physically demanding was the task? *scale: best-to-worst* |
| **Correlation:** | 0.537 |
| **Sig. (2-tailed):** | 0.002 |

Table 5.4: Significant correlation between questions about HMD quality and physical demand

| | |
|---|---|
| **Question SC8:** | How much did the visual display quality (HMD) interfere or distract you from performing the assigned tasks? *scale: best-to-worst* |
| **Question TLX6:** | How insecure, discouraged, irritated, stressed, and annoyed were you? *scale: best-to-worst* |
| **Correlation:** | 0.537 |
| **Sig. (2-tailed):** | 0.002 |

Table 5.5: Significant correlation between questions about HMD quality and frustration

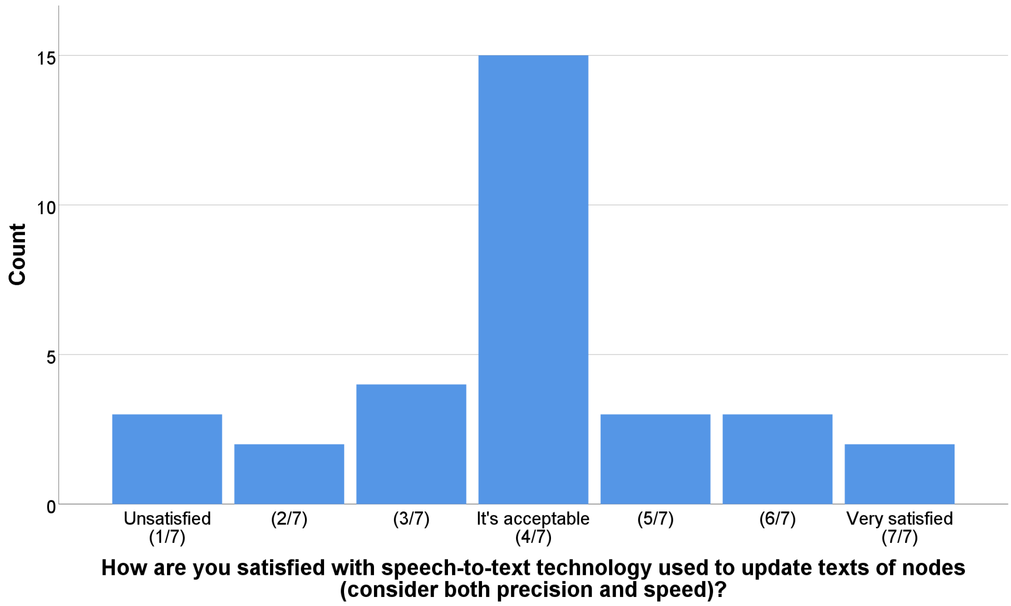


Figure 5.6: Satisfaction with speech recognition

| | |
|---|---|
| **Question SC13.5:** | Evaluation of usability of "text change" feature *scale: worst-to-best* |
| **Question TLX6:** | How insecure, discouraged, irritated, stressed, and annoyed were you? *scale: best-to-worst* |
| **Correlation:** | -0.481 |
| **Sig. (2-tailed):** | 0.005 |

Table 5.6: Significant correlation between questions about usability of "text change" feature and frustration

### 5.3.3 Log Results

The activity of participants during the testing scenario was logged in order to get more information about their behavior as well as the effectiveness of our platform. Stored information contains general actions performed by the user, like node creation and deletion, and visualizations of mind map canvas interactions. These visualizations are of two types, and both are in the form of a heatmap, i.e., graphical representation of particular behavior. Figure 5.7 is created by merging heatmaps of all tested users. The points in the image represent positions in the mind map canvas which were "hit" by a laser pointer while aiming at the canvas and selecting. The RGB colors determine the relative amount of hits at a given pixel with red being the most "hit" pixels and blue being the least "hit" pixels, whereas green pixels are somewhere in between. Figure 5.8 shows an averaged heatmap of selected nodes of all users. In other words, this picture determines positions where nodes were selected for the longest time. In this case holds that the bigger the opacity is, the longer this position was covered by some node in a selected state.

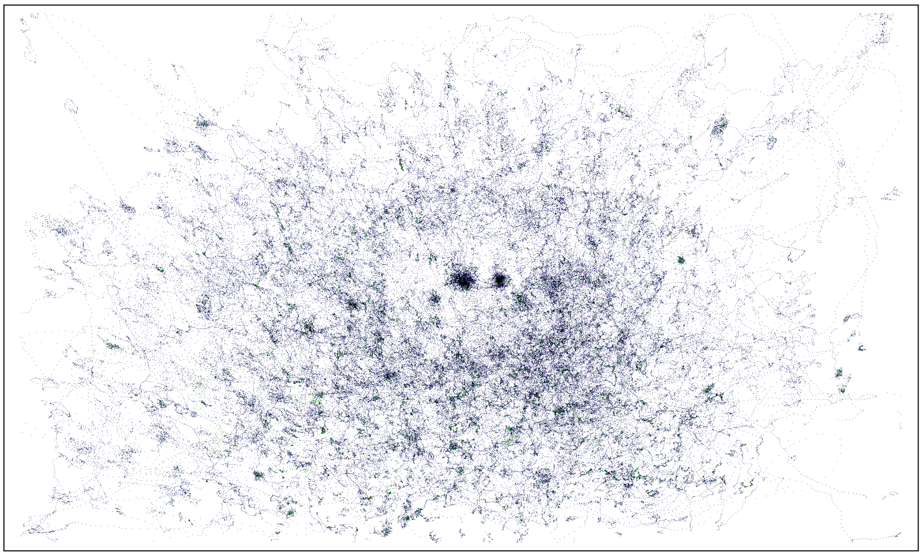

Figure 5.7: Merged heatmap with pointer movements of all users

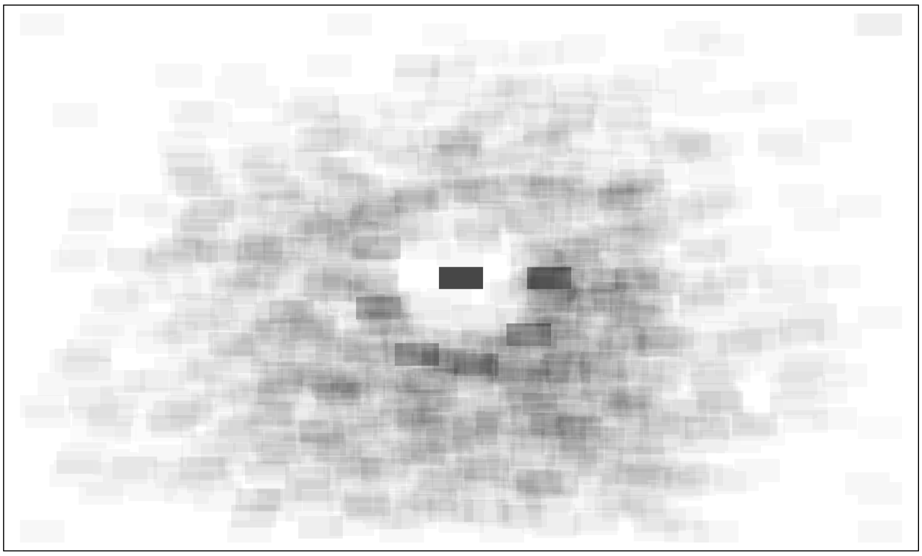

Figure 5.8: Merged heatmap highlighting positions of selected nodes

The first interesting observation regarding both heatmaps is the fact that the space near the corners of the mind map is unused. This shows the tendency of users to place most of the nodes near the central root node. Another interesting observation is a rather significant difference in the density of heatmap in the bottom and the upper half of the canvas. This confirms that there might be reduced readability in the upper half and users are therefore preferring nodes which are closer to them, i.e., at the bottom of the canvas. Figure 5.8 also reveals that users generally like to move the nodes around as they wish and they do not just stick to the default automatic circular placement. This means that it is necessary to have a good interface for node movement. Regarding the movement, in order to be more precise, Figures 5.9 and 5.10 show two heatmaps. If examined in more detail, it is apparent that there are noticeable differences in the overall look – the reason is that there are two types of users. The first type is less common and prefers to stick to the default node placement and does only minor changes while the second category of users is more active in this regard. This is also related to another observed user behavior – some people use the laser pointer nearly all the time while others use it only when necessary. The former group of users will probably appreciate the updated controls scheme in the latest version of the system, as described in section 4.3, because it will not force them to press the button all the time.

Figure 5.9: Example heatmap of the first type of users reorganizing nodes rather rarely (image is rotated by 90 degrees for readability)

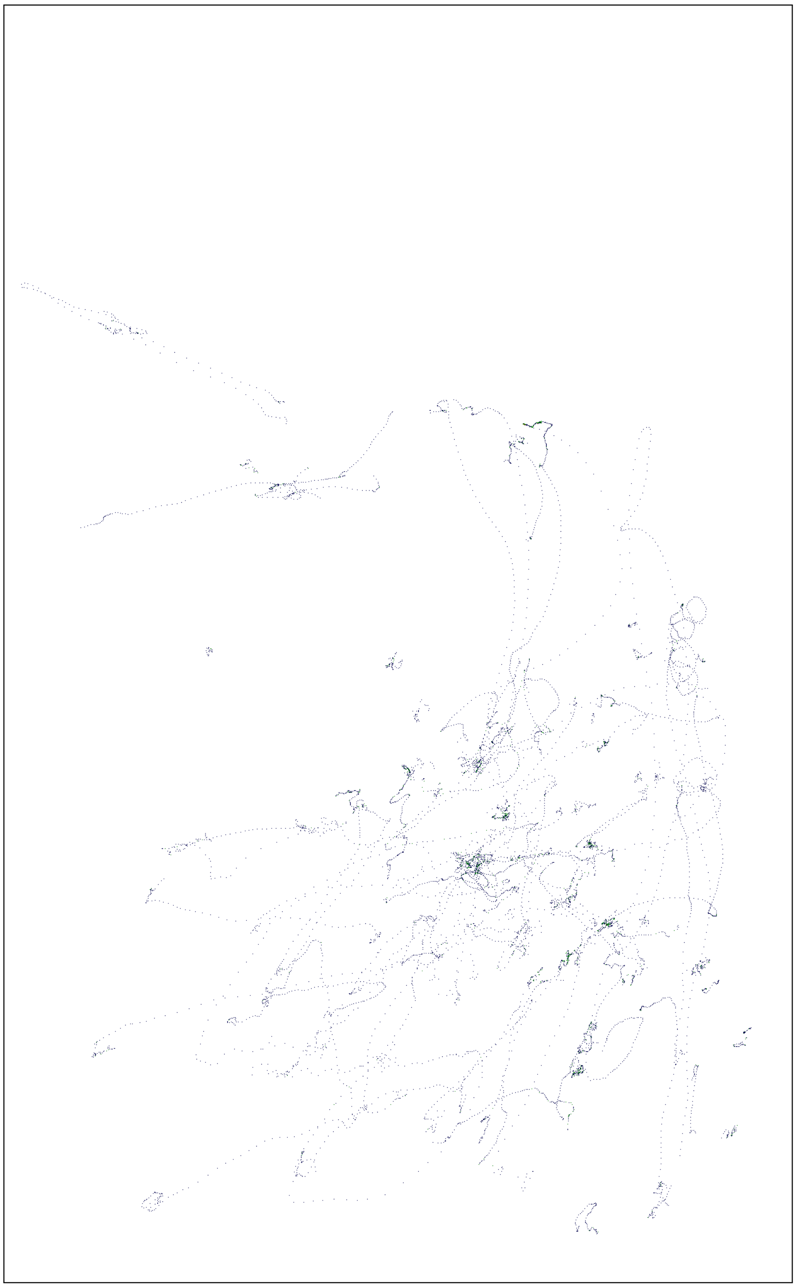

Figure 5.10: Example heatmap of the second type of users with more active mind map reorganization (image is rotated by 90 degrees for readability)

# 6 Conclusion And Future Work

This chapter consists of two parts. The first one reviews the thesis, its objectives, execution, and results. The second part presents ideas for future work.

## 6.1 Discussion

This thesis presented a multimodal virtual reality mind map application allowing several participants to fully experience the process of brainstorming despite their physical distance. Provided functionality covers both the idea-generation phase as well as voting about the best ideas. Multimodality is achieved thanks to the combination of speech and controller. During any time of the session, it is possible to export the current mind map to XMind format. The system is designed to work with HTC Vive (Pro) headset.

To verify the usability of the system, experiment with 32 participants (19 male, 13 female) was conducted. Users were tested in pairs and filled several questionnaires summarizing their experience. The results indicate that the system performs according to its specifications and does not show critical problems. However, there is a space for improvement. Some problems from the users' feedback were already fixed or improved in the latest version. Undo/redo feature was added and the whole environment was redesigned with mind map canvas positioned lower. Controls scheme was also simplified so, at the moment, only one button is used. It is not clear whether this input device is the best solution though. It has several positives; the most important one is reliability – it works all the time without problems. It is clear whether the user presses a button or not. Also, it is widely supported by development kits. The negatives include the fact that this kind of interaction is not really natural; an ability to control the system with bare hands would be more realistic. Nevertheless, this would bring new problems related to the reliability. Besides that, to justify these new challenges, it would be necessary to find gestures which are both natural and usable at the same time.

However, the biggest question, which is not yet fully answered, is whether the advantages of virtual reality overweight the disadvantages. In other words, whether it is necessary to have such an application in the virtual reality or not. The key benefits of this solution are immersion, fun, and reduction in outside distractions. These factors can have a positive influence on the outputs of the brainstorming session. On the other hand, the negatives include increased physical demand, cost, equipment overhead, possible problems with readability, and less reliable and slower means of input. Especially the last point is important because the brainstorming effectivity would be diminished if the users would not be able to place at the canvas whatever ideas they want because the speech recognition is having problems with given words. Therefore, to justify the usage of virtual reality, the positive impacts should be maximized, and the negatives should be suppressed as much as possible. Since the immersion is the strongest point of virtual reality, the goal is to maximize it. What can help is the improvement of user avatars and the creation of several new environments. Then, users would be able to choose whatever environment suits them most and helps them to feel comfortable so they can produce as many ideas as possible. The problems related to input can be partially solved by having a better speech recognition software while also offering some additional input options like virtual keyboard or list of related words to the one just said. This idea can be extended even further. Every time a users says a label for a node, the system might automatically place some related texts or images into the environment. The user will be, therefore, literally surrounded by ideas, which will provide him or her hints on how to proceed. The environment will then become a constantly evolving source of inspiration.

Another question is what additional options and tools the environment can offer to increase the effectivity of the brainstorming session. Some desktop sharing feature, or a web browser, might help to move the experience to the new levels. However, it is important to keep in mind that new tools should improve effectivity and not serve as distractions.

## 6.2 Future Work

There are many possibilities on how to improve and extend the system; some of them were already mentioned in the text. Besides general improvements to the interface, avatars will be exchanged for some more realistic ones. Also, name tags will be added to identify individual participants. Thanks to the integrated voice solution, some speech-related features will be added, for example, automatic muting of users when they are saying a label for a node. Moreover, there is also going to be visual feedback, like icon or mouth animation, to make it clear which user is speaking. Possibilities of hand gesture controls will be examined as well. In the end, a comparative user study will be made between traditional platforms for remote collaboration and the virtual reality mind map to assess the advantages and disadvantages of each approach.

# A An Appendix

Following appendices of this master's thesis can be found in the archive of the Information System of Masaryk University:

- Source code of the application
- Two builds of the application
  - User study version
  - Latest version
- Questionnaires utilized during the user study
- Example video showing the system in progress

# Bibliography


1. OSSIC. *The History of Virtual Reality* [online]. OSSIC Blog, 2017 [visited on 2019-05-12]. Available from: https://medium.com/ossic/the-history-of-virtual-reality-e517983b1253.
2. MOON, Mariella. *Ford is letting some designers build cars in virtual reality* [online]. 2019 [visited on 2019-05-12]. Available from: https://www.engadget.com/2019/01/29/ford-gravity-sketch/.
3. VELICHKO, Michael. *VR military training - the next step of combat evolution* [online]. Jasoren, 2018 [visited on 2019-05-12]. Available from: https://jasoren.com/vr-military-training-the-next-step-of-combat-evolution/.
4. *Virtual reality headsets* [online]. 2016 [visited on 2019-05-12]. Available from: https://www.hypergridbusiness.com/faq/best-virtual-reality-headsets/.
5. *Augmented Reality and Virtual Reality Headsets Poised for Significant Growth, According to IDC* [online]. 2019 [visited on 2019-05-12]. Available from: https://www.idc.com/getdoc.jsp?containerId=prUS44966319.
6. *World Internet Users Statistics and 2019 World Population Stats* [online]. Miniwatts Marketing Group [visited on 2019-05-12]. Available from: https://www.internetworldstats.com/stats.htm.
7. *Speedtest Global Index – Monthly comparisons of internet speeds from around the world* [online] [visited on 2019-05-12]. Available from: https://www.speedtest.net/global-index.
8. HICKS, Michael. *The best multiplayer VR games* [online]. TechRadar The source for Tech Buying Advice, 2018 [visited on 2019-05-12]. Available from: https://www.techradar.com/news/the-best-multiplayer-vr-games.
9. *rumii | Virtual Reality Conferencing and Collaboration Software* [online] [visited on 2019-03-22]. Available from: https://www.rumii.net/.
10. GRUBB, Roy. *Roots of visual mapping* [online]. The mind-mapping.org Blog [visited on 2019-03-04]. Available from: https://www.mind-mapping.org/blog/roots-of-visual-mapping/.

11. FARRAND, Paul; HUSSAIN, Fearzana; HENNESSY, Enid. The efficacy of the 'mind map' study technique. *Medical education*. 2002, vol. 36, pp. 426–31. Available from DOI: 10.1046/j.1365-2923.2002.01205.x.

12. *XMind official website* [online] [visited on 2019-03-04]. Available from: https://www.xmind.net/.

13. BUZAN, Tony. *Use your head*. Guild Publishing & BBC Publications, 1984.

14. BUZAN, Tony; BUZAN, Barry. *The mind map book: how to use radiant thinking to maximize your brains untapped potential*. Plume, 1996.

15. BUZAN, Tony; ABBOTT, Susanna. *The ultimate book of mind maps: unlock your creativity, boost your memory, change your life*. Thorsons, 2006.

16. MADU, B.C.; METU IFEOMA, C. Effect of mind map as a note-taking approach on students' achievements in Economics. *Journal of Emerging Trends in Economics and Management Sciences*. 2012, vol. 3, pp. 247–251. Available also from: https://journals.co.za/content/sl_jetems/3/3/EJC132206.

17. ROODA, Linda. Effects of Mind Mapping on Student Achievement in a Nursing Research Course. *Nurse educator*. 1994, vol. 19, pp. 25–7. Available from DOI: 10.1097/00006223-199411000-00015.

18. BURGESS-ALLEN, Jilla; OWEN-SMITH, Vicci. Using mind mapping techniques for rapid qualitative data analysis in public participation processes. *Health Expectations*. 2010, vol. 13, no. 4, pp. 406–415. Available from DOI: 10.1111/j.1369-7625.2010.00594.x.

19. KOTOB, F.; STYGER, L.; RICHARDSON, L. P. Exploring mind mapping techniques to analyse complex case study data. *Australian Academy of Business and Economics Review*. 2016, vol. 2, pp. 244–262.

20. VIJAYAVALSALAN, B. Mind mapping as a strategy for enhancing essay writing skills. 2016, vol. 45, pp. 137–150. Available from DOI: 10.15804/tner.2016.45.3.11.

21. CONRAD, B. Evolution of the brainstorming and planning. *PMI Global Congress Proceedings* [online]. 2009 [visited on 2019-03-05]. Available from: https://www.pmi.org/learning/library/evolution-brainstorming-planning-mindmapping-6770.

22. ABD KARIM, Rafidah; ABU, Abdul. BRAINSTORMING APPROACH AND MIND MAPPING IN WRITING ACTIVITY. *1st English Education International Conference (EEIC). Unsyiah, Banda Aceh* [online]. 2016 [visited on 2019-03-05]. Available from: `https://www.researchgate.net/publication/316138865_BRAINSTORMING_APPROACH_AND_MIND_MAPPING_IN_WRITING_ACTIVITY`.

23. OSBORN, Alex F. *How to think up*. McGraw-Hill, 1942.

24. BASS/PFEIFFER, Jossey. *Brainstorming* [online]. The Pfeiffer Library Volume 26, 2nd Edition, 1998 [visited on 2019-03-07]. Available from: `http://home.snu.edu/~jsmith/library/body/v26.pdf`.

25. ADAMS, James L. *Conceptual blockbusting: a guide to better ideas*. 4th ed. Perseus Pub., 2001.

26. *VIRTUAL REALITY | meaning in the Cambridge English Dictionary* [online]. Cambridge University Press [visited on 2019-03-11]. Available from: `https://dictionary.cambridge.org/dictionary/english/virtual-reality`.

27. JERALD, Jason. *The VR book: human-centered design for virtual reality*. Association for Computing Machinery, 2016.

28. BOWMAN, D. A.; MCMAHAN, R. P. Virtual Reality: How Much Immersion Is Enough? *Computer*. 2007, vol. 40, no. 7, pp. 36–43. ISSN 0018-9162. Available from DOI: `10.1109/MC.2007.257`.

29. JENNETT, Ch.; COX, A. L.; CAIRNS, P.; DHOPAREE, S.; EPPS, A.; TIJS, T.; WALTON, A. Measuring and defining the experience of immersion in games. *International Journal of Human-Computer Studies*. 2008, vol. 66, no. 9, pp. 641–661. ISSN 1071-5819. Available from DOI: `https://doi.org/10.1016/j.ijhcs.2008.04.004`.

30. *VIRTUAL PARACHUTE TRAINING* [online]. Bohemia Interactive Simulations [visited on 2019-03-13]. Available from: `https://bisimulations.com/company/customer-showcase/virtual-parachute-training`.

31. *Virtual Reality on Steam* [online]. Valve Corporation [visited on 2019-03-12]. Available from: `https://store.steampowered.com/vr/`.

32. DE LOOPER, CH. *The best VR movies: experience the next era of storytelling* [online]. Techradar, 2018 [visited on 2019-03-12]. Available from: https://www.techradar.com/news/the-best-vr-movies-experience-the-next-era-of-storytelling.

33. *Virtual Reality (VR) - Statistics & Facts* [online]. Statista [visited on 2019-03-12]. Available from: https://www.statista.com/topics/2532/virtual-reality-vr.

34. *VR-BASED GUNSHIP CREW TRAINER* [online]. Bohemia Interactive Simulations [visited on 2019-03-13]. Available from: https://bisimulations.com/company/customer-showcase/vr-based-gunship-crew-trainer.

35. FACTORS, RTO Human; (HFM), Medicine Panel. *Virtual Reality: State of Military Research and Applications in Member Countries* [online]. NATO's Research and Technology Organisation [visited on 2019-03-13]. ISBN 92-837-0030-9. Available from: https://apps.dtic.mil/dtic/tr/fulltext/u2/a411978.pdf.

36. HAAR, René ter. Virtual Reality in the Military : Present and Future. In: 2005. Available also from: https://pdfs.semanticscholar.org/8076/c5eeb49d9a6e7d4eb603681083566e1dd958.pdf.

37. PANTELIDIS, Panteleimon; CHORTI, Angeliki; PAPAGIOUVANNI, Ioanna; PAPAROIDAMIS, Georgios; DROSOS, Christos; PANAGIOTAKOPOULOS, Thrasyvoulos; LALES, Georgios; SIDERIS, Michail. Virtual and Augmented Reality in Medical Education. In: TSOULFAS, Georgios (ed.). *Medical and Surgical Education*. Rijeka: IntechOpen, 2018, chap. 5. Available from DOI: 10.5772/intechopen.71963.

38. SAMADBEIK, Mahnaz; YAAGHOBI, Donya; BASTANI, Peivand; ABHARI, Shahabeddin; REZAEE, Rita; GARAVAND, Ali. *The Applications of Virtual Reality Technology in Medical Groups Teaching*. Journal of Advances in Medical Education & Professionalism, 2018. Available also from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6039818/.

39. MORITZ, Elke; MEYER, Joerg. Interactive 3D protein structure visualization using virtual reality. In: *Proceedings - Fourth IEEE Symposium on Bioinformatics and Bioengineering, BIBE*. 2004, pp. 503–507. ISBN 0-7695-2173-8. Available from DOI: 10.1109/BIBE.2004.1317384.

40. *Google Expeditions* [online]. Google [visited on 2019-03-16]. Available from: https://edu.google.com/products/vr-ar/expeditions.

41. *IrisVR* [online]. IrisVR [visited on 2019-03-16]. Available from: https://irisvr.com/.

42. *MeetinVR* [online]. MeetinVR [visited on 2019-03-16]. Available from: https://meetinvr.net/.

43. *Google Trends – Virtual Reality* [online]. Google [visited on 2019-03-17]. Available from: https://trends.google.com/trends/explore?date=all&q=%2Fm%2F07_ny.

44. *History Of Virtual Reality* [online]. Virtual Reality Society [visited on 2019-03-18]. Available from: https://www.vrs.org.uk/virtual-reality/history.html.

45. SHARMISTHA, M. Brief Introduction of Virtual Reality & its Challenges. In: *International Journal of Scientific & Engineering Research*. 2013, vol. 4, pp. 304–309. ISSN 2229-5518.

46. GRIERSON, T. *8 Things You Need to Know About the 4DX Theater Experience* [online]. Rolling Stone, LLC, 2014 [visited on 2019-03-18]. Available from: https://www.rollingstone.com/culture/culture-news/8-things-you-need-to-know-about-the-4dx-theater-experience-86801/.

47. BARNARD, D. *History of VR - Timeline of Events and Tech Development* [online]. VirtualSpeech Ltd., 2018 [visited on 2019-03-18]. Available from: https://virtualspeech.com/blog/history-of-vr.

48. IVROX. *A Brief History of Virtual Reality* [online]. VRFocus, 2018 [visited on 2019-03-18]. Available from: https://www.vrfocus.com/2018/05/a-brief-history-of-virtual-reality/.

49. BROWN, L. *A Brief History of Virtual Reality* [online]. Wondershare, 2017 [visited on 2019-03-18]. Available from: https://filmora.wondershare.com/virtual-reality/history-of-vr.html.

50. HAVIG, Paul; MCINTIRE, John; GEISELMAN, Eric. Virtual reality in a cave: Limitations and the need for HMDs? *Proceedings of SPIE - The International Society for Optical Engineering*. 2011, vol. 8041. Available from DOI: 10.1117/12.883855.

51. CHERDO, L. *VR headset buying guide: how to choose a virtual reality HMD?* [online]. Aniwaa Pte. Ltd., 2018 [visited on 2019-03-18]. Available from: `https://www.aniwaa.com/guide/vr-ar/vr-headset-buying-guide/`.

52. *Unit shipments of virtual reality (VR) devices worldwide from 2017 to 2019 (in millions), by vendor* [online]. Statista, 2018 [visited on 2019-03-18]. Available from: `https://www.statista.com/statistics/671403/global-virtual-reality-device-shipments-by-vendor/`.

53. LAMKIN, P. *Best VR headsets 2019: HTC Vive, Oculus, PlayStation VR compared* [online]. Wareable Ltd., 2019 [visited on 2019-03-18]. Available from: `https://www.wareable.com/vr/best-vr-headsets-2017`.

54. KHODAEI, S. *VR has disappeared from the Gartner hype cycle 2018* [online]. vr-on GmbH, 2018 [visited on 2019-03-18]. Available from: `http://www.vr-on.com/blog-detail-2084/vr-has-disappeared-from-the-gartner-hype-cycle-201.html`.

55. *MODALITY | meaning in the Cambridge English Dictionary* [online]. Cambridge University Press [visited on 2019-03-19]. Available from: `https://dictionary.cambridge.org/dictionary/english/modality`.

56. AMMA, Ch.; PUTZE, F. *Multimodal Interaction [presentation]* [online]. Karlsruhe Institute of Technology, 2012 [visited on 2019-03-19]. Available from: `https://csl.anthropomatik.kit.edu/downloads/vorlesungsinhalte/DEIB1213_V02-03_Multimodal_Interaction.pdf`.

57. DUMAS, Bruno; LALANNE, Denis; OVIATT, Sharon L. Multimodal Interfaces: A Survey of Principles, Models and Frameworks. In: *Human Machine Interaction*. 2009.

58. NIGAY, Laurence; COUTAZ, Joëlle. A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion. In: *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*. Amsterdam, The Netherlands: ACM, 1993, pp. 172–178. CHI '93. ISBN 0-89791-575-5. Available from DOI: `10.1145/169059.169143`.

59. BOLT, Richard A. "Put-that-there": Voice and Gesture at the Graphics Interface. *SIGGRAPH Comput. Graph.* 1980, vol. 14, no. 3, pp. 262–270. ISSN 0097-8930. Available from DOI: 10.1145/965105.807503.

60. JACKO, J.A. *Human Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications, Third Edition*. CRC Press, 2012. Human Factors and Ergonomics. ISBN 9781439829448. Available also from: https://books.google.cz/books?id=dVrRBQAAQBAJ.

61. MAYBURY, Mark T.; WAHLSTER, Wolfgang. *Readings in intelligent user interfaces*. Morgan Kaufmann, 1998.

62. JOKINEN, Kristiina; RAIKE, Antti. Multimodality - Technology, Visions and Demands for the Future. In: *In: Proceedings of the 1 st Nordic Symposium on Multimodal Interfaces*. 2003.

63. STEPHANIDIS, C. *The Universal Access Handbook*. CRC Press, 2009. Human Factors and Ergonomics. ISBN 9781420064995. Available also from: https://books.google.cz/books?id=Nrti8Q7K1wwC.

64. KATSAMANIS, Athanasios; PITSIKALIS, Vassilis; THEODORAKIS, Stavros; MARAGOS, Petros. The Handbook of Multimodal-Multisensor Interfaces. In: OVIATT, Sharon; SCHULLER, Björn; COHEN, Philip R.; SONNTAG, Daniel; POTAMIANOS, Gerasimos; KRÜGER, Antonio (eds.). New York, NY, USA: Association for Computing Machinery and Morgan & Claypool, 2017, chap. Multimodal Gesture Recognition, pp. 449–487. ISBN 978-1-97000-167-9. Available from DOI: 10.1145/3015783.3015796.

65. KARPOV, A. A. An automatic multimodal speech recognition system with audio and video information. *Automation and Remote Control*. 2014, vol. 75, no. 12, pp. 2190–2200. ISSN 1608-3032. Available from DOI: 10.1134/S000511791412008X.

66. GÜRKÖK, Hayrettin; NIJHOLT, Anton. Brain–Computer Interfaces for Multimodal Interaction: A Survey and Principles. *International Journal of Human–Computer Interaction*. 2012, vol. 28, no. 5, pp. 292–307. Available from DOI: 10.1080/10447318.2011.582022.

67. AWADA, I. A.; MOCANU, I.; FLOREA, A. M.; CRAMARIUC, B. Multimodal Interface for Elderly People. In: *2017 21st International Conference on Control Systems and Computer Science (CSCS)*. 2017, pp. 536–541. ISSN 2379-0482. Available from DOI: 10.1109/CSCS.2017.82.

68. RODRIGUES, Élvio; CARREIRA, Micael; GONÇALVES, Daniel. Developing a Multimodal Interface for the Elderly. *Procedia Computer Science*. 2014, vol. 27, pp. 359–368. Available from DOI: 10.1016/j.procs.2014.02.040.

69. CHHABRIA, Sharda; DHARASKAR, Rajiv. Multimodal interface for disabled persons. *International Journal of Computer Science and Communication*. 2011, vol. 2, pp. 223–228.

70. KARPOV, Alexey; RONZHIN, Andrey. ICANDO: Low cost multimodal interface for hand disabled people. *Journal on Multimodal User Interfaces*. 2007, vol. 1, no. 2, pp. 21–29. ISSN 1783-8738. Available from DOI: 10.1007/BF02910056.

71. CHERNAKOVA, Svetlana; NECHAEV, Alexander; KARPOV, Alexey; RONZHIN, Andrey. Assistive Multimodal Interface for Medical Applications. *SPECOM 2006*. 2006.

72. DI FIORE, Fabian; VANDOREN, Peter; VAN REETH, Frank. Multimodal Interaction in a Collaborative Virtual Brainstorming Environment. In: 2004, vol. 3190, pp. 47–60. Available from DOI: 10.1007/978-3-540-30103-5_6.

73. *iMindMap Official Website* [online] [visited on 2019-03-21]. Available from: https://imindmap.com/.

74. *MindMeister Official Website* [online] [visited on 2019-03-21]. Available from: https://www.mindmeister.com.

75. *Noda on Steam* [online]. Coding Leap [visited on 2019-03-21]. Available from: https://store.steampowered.com/app/578060/Noda/.

76. *Mind Map VR on Steam* [online]. VR-AR Corp [visited on 2019-04-01]. Available from: https://store.steampowered.com/app/885250/Mind_Map_VR__VR/.

77. MIYASUGI, Masayuki; AKAIKE, Hideo; NAKAYAMA, Yasuichi; KAKUDA, Hiroyasu. Implementation and evaluation of multi-user mind map authoring system using virtual reality and hand gestures. *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*. 2017. Available from DOI: 10.1109/gcce.2017.8229432.

78. BARYN. *Deep Analysis of the Vive Controller in light of the Steam Controller* [online]. Reddit [visited on 2019-04-01]. Available from: https://www.reddit.com/r/SteamController/comments/4ilxwz/deep_analysis_of_the_vive_controller_in_light_of/?utm_source=share&utm_medium=web2x.

79. *Leap Motion* [online]. LEAP MOTION, INC. [visited on 2019-04-20]. Available from: https://www.leapmotion.com/.

80. *Unity Official Website* [online]. Unity Technologies [visited on 2019-04-04]. Available from: https://unity.com/.

81. *Unity Public Relations* [online]. Unity Technologies [visited on 2019-04-04]. Available from: https://unity3d.com/public-relations.

82. *SteamVR* [online]. Valve Corporation [visited on 2019-04-04]. Available from: http://steamvr.com.

83. *VRTK – Virtual Reality Toolkit* [online]. thestonefox [visited on 2019-04-04]. Available from: https://vrtoolkit.readme.io/.

84. *CMUSphinx Open Source Speech Recognition* [online]. Carnegie Mellon University [visited on 2019-04-26]. Available from: https://cmusphinx.github.io/.

85. *Wit.ai* [online]. Wit.ai, Inc. [visited on 2019-04-26]. Available from: https://wit.ai/.

86. DOLEZAL, Milan; CHMELIK, Jiri; LIAROKAPIS, Fotis. An immersive virtual environment for collaborative geovisualization. *2017 9th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games)*. 2017. Available from DOI: 10.1109/vs-games.2017.8056613.

87. *UNet Deprecation FAQ* [online]. Unity [visited on 2019-04-10]. Available from: https://support.unity3d.com/hc/en-us/articles/360001252086-UNet-Deprecation-FAQ.

88. *ZeroTier | Home* [online]. ZeroTier Inc. [visited on 2019-04-10]. Available from: `https://www.zerotier.com/`.

89. LIU, Yifan. *Is VR True to Scale? Experiments Say Yes* [online]. IrisVR Blog, 2015 [visited on 2019-04-15]. Available from: `https://blog.irisvr.com/blog/category/is-vr-true-to-scale-two-scientific-experiments`.

90. *Agora.io Real-Time Voice and Video Engagement* [online] [visited on 2019-05-03]. Available from: `https://www.agora.io`.

91. WITMER, Bob G.; SINGER, Michael J. Measuring Presence in Virtual Environments: A Presence Questionnaire. *Presence: Teleoperators and Virtual Environments*. 1998, vol. 7, no. 3, pp. 225–240. Available from DOI: `10.1162/105474698565686`.

92. WITMER, Bob G.; JEROME, Christian J.; SINGER, Michael J. The Factor Structure of the Presence Questionnaire. *Presence: Teleoperators and Virtual Environments*. 2005, vol. 14, no. 3, pp. 298–312. Available from DOI: `10.1162/105474605323384654`.

93. HART, Sandra G. Nasa-task load index (NASA-TLX); 20 years later. *PsycEXTRA Dataset*. 2006. Available from DOI: `10.1037/e577632012-009`.