Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Redesigning the 2D Paradigm of Computer Work for Head Mounted Displays

Liam Byrne

August, 2023

A Dissertation submitted in partial fulfilment

of the requirements for the degree of

Master of Science in Computer Science (Augmented and Virtual Reality)

Video Demonstration

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Liam Byrne

August 18th, 2023

Table of Contents

List of Figures

Abstract

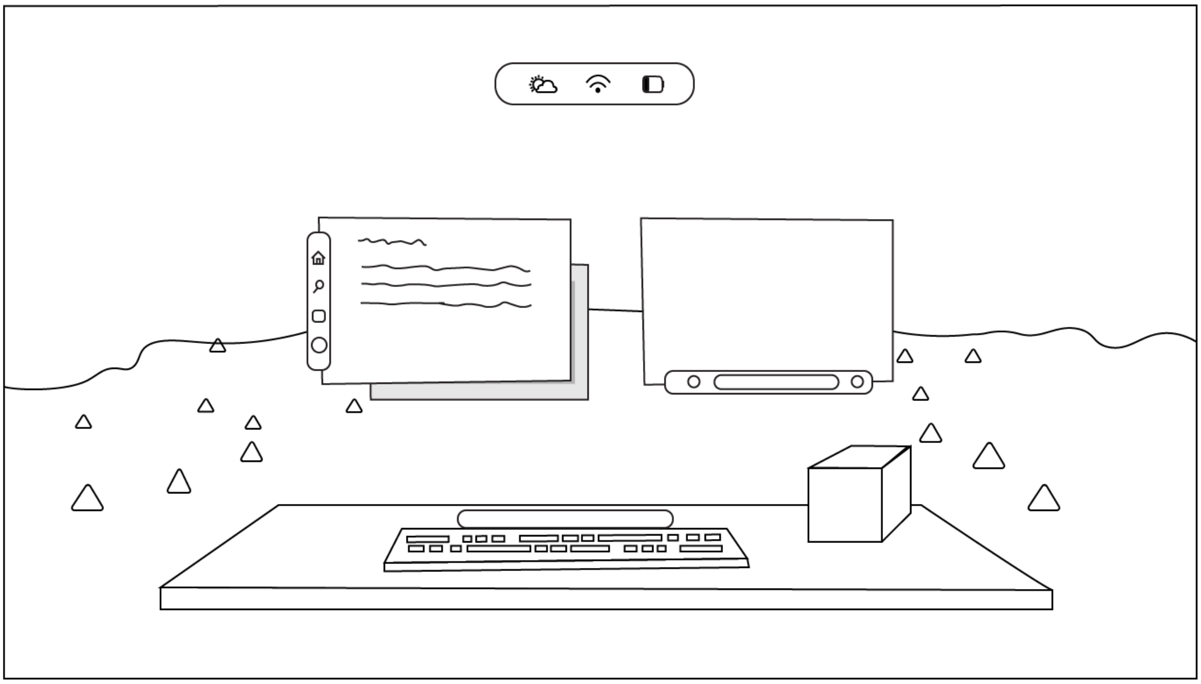

Figure 1. A mockup of a computer work application with windows, a volume and a virtual environment

The rise of HMDs brings forth possibilities for reimagining computer work in more immersive and productive ways indicating a shift in how humans interact with computers. HMDs and spatial interfaces are progressing towards becoming an alternative to two dimensional computer interactions. Designing mixed reality (MR) interfaces, which are more complex compared to desktop interfaces, requires standards for creating applications especially when leveraging the spatial capabilities of HMDs. Before HMDs can be widely adopted for tasks the tools currently used in a two-dimensional context will need to be redesigned to take advantage of the third dimension depth. This paper offers design guidelines for developing computer work applications for HMDs. These guidelines along with a methodology for redesigning 2D computer work for virtual environments are demonstrated through a sample application that explores how MR can enhance the research process. Findings from the design and implementation of this project, such as areas to focus the redesign around, are then shared to benefit future projects in the rapidly evolving field of interface design for HMDs.

Acknowledgements

I would like to thank my supervisor Dr. Gavin Doherty for his guidance and support over the course of my dissertation.

I would also like to thank Rob Bowman for providing feedback and advice throughout the project.

A special thanks to Cormac Madden for generously providing the headset essential for this research.

Lastly, I am grateful to my friends and family for their continued support throughout this process.

1. Introduction

New possibilities for human-computer interaction have been created by the introduction of HMDs, including augmented and virtual reality devices. While current usage of MR has been primarily content consumption, there is untapped potential for more productive and creative use. There is now a growing need to explore and expand the possibilities offered by HMDs spurred by technology advances and investment interest in MR. A promising area, which has the potential to revolutionise how we work with computers, is redesigning the 2D paradigm of computer work for the use of HMDs. More immersive and efficient work environments could thus be created that will ultimately enhance productivity and user experience.

Digital interfaces that allow people to use their bodies and the space in which they exist to complete tasks have belonged thus far to the realm of science fiction, but efforts are now being made to bring this technology to life, providing users with a more immersive and instinctive way of interacting with computer systems. HMDs and spatial interfaces could soon be widely used in the workplace as a valid alternative or addition to conventional two-dimensional computer-based work.

The limited adoption of MR applications can be attributed, in part, to the difficulties involved in developing applications. Specifically the design of interfaces for MR applications is more intricate and demanding compared to interfaces, for desktop based applications.

In an environment where HMDs become the primary way to interact with your computer or the computer itself, new standards will need to emerge for creating applications. How would a designer go about designing or more likely redesigning their application for this new medium? While conventional interfaces mainly use 2D graphical displays, MR interfaces use both 2D and 3D displays. How can they effectively utilise the spatial capabilities of HMDs to create more intuitive and immersive user experiences than with traditional 2D displays?

Given that HMD technology is constantly improving and is gaining increased attention from investors and developers, as a new computing paradigm, it seems logical that increasing thought should be put into how to design for these new interfaces. It seems particularly important to consider interface design in a working environment as the average knowledge worker spends a significant amount of time in front of a computer screen and relies heavily on the interface to perform their tasks efficiently and effectively. This project will focus on how interfaces should be designed for HMDs to best take advantage of the added z dimension (depth) which they enable.

Contemplating this, the following query was posed: Is it possible to reimagine the conventional two-dimensional model of computer work for use with HMDs? This inquiry presents an intriguing aspect—design. The concept of design suggests a systematic approach to redesigning computer work for MR, considering various factors that influence the process.

Before this shift in the computing paradigm can occur, fundamental questions need to be addressed and challenges overcome. Only then could we transition smoothly to the implementation of HMDs in creative and productive applications. Naturally, the tools that are designed for traditional 2D screens, are not tailored to the unique demands and capabilities of HMDs. As an example, until recently, users couldn't see their physical keyboard while working in an immersive VR environment, making text entry and data input very difficult. Integrating and allowing for the interaction of two-dimensional content in a three-dimensional environment presents further challenges.

This relatively new field of interface design for HMDs necessitates the discovery of design principles and their implementation. This rapidly evolving field requires a well-rounded research approach in order to remain current. This paper outlines design guidelines for HMDs work applications and presents a demo application that showcases these guidelines in action, accompanied by anecdotal findings regarding the design and implementation of the system, which will hopefully help inform future work in this field.

The design guidelines and methodology in this report focus on adapting 2D screens for spatial computing by a) projecting them into 3D space within a custom virtual environment and b) augmenting the experience with a volumetric interface. The primary benefit of adopting this methodology is its ability to adapt existing 2D tools for use with HMDs while leveraging the unique benefits of 3D visualisation and interaction. This means that rather than starting from scratch, developers and designers can build upon existing 2D tools and workflows, enhancing them with spatial computing capabilities. This approach allows for a more seamless transition for users accustomed to traditional 2D interfaces and reduces the learning curve associated with adopting a completely new operating system.

In the first half of the report how to redesign work applications for HMDs based on research is discussed. In the second half, these research findings are applied and in evaluating the resulting application, foundational knowledge for future endeavours aimed at creating VR tools tailored for computer work is gained.

2. Background on MR

This section aims to provide a brief overview of the core concepts related to the topic of interface design for HMDs. In this section explain related and often confused terms, headsets and 3D GUIs.

Related Terms

Head Mounted Displays (HMD's) are available on the market and becoming more popular however there remains a degree of confusion between key terms like Augmented Reality (AR) and Virtual Reality (VR) and the continuum between them which Milgram et al coined as Mixed Reality (MR). In their seminal paper on the Reality-Virtuality Continuum.

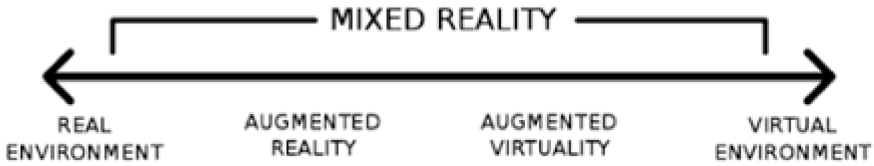

Figure 4. The Reality-Virtuality Continuum by Milgram et al

Jaron Lanier first coined the term, VR (Lanier & Biocca, 1992). Since then multiple definitions have been proposed. Sherman and Craig concisely defined VR using four criteria 1) a virtual world 2) immersion, 3) sensory feedback, and 4) interactivity. Some researchers choose to refer to virtual environments (VE) as opposed to VR, due to beliefs that the hype around the term VR has resulted in inaccurate understanding and expectations. (Bowman et al 2004, Earnshaw et al., 1993). The term immersion is often used concerning VR to signify the sense of involvement in an experience in which your senses are substituted with information. (Sherman and Craig, 2003). The extent to which your brain accepts the sensory information it's given as reality is called "Presence".

AR is closely related to VR. AR experiences enhance the physical world with virtual overlays (Sherman and Craig, 2003, Adam, 1993) AR experiences include both the virtual and real world by overlaying computer-generated imagery onto the user's perception of the real world. The phrase XR comes from the Chronos Group combining the V of (Virtual Reality) and the A of (Augmented Reality) to make an X for Extended Reality. Increasingly digital content exists on a continuum from fully physical to fully virtual. Rather than refer to any point on the continuum explicitly, This report refers to MR or Mixed Reality regularly as this paper discusses the design implications of HMDs and virtual environments which are generally applicable to both AR and VR applications.

3D GUIs

Graphical User Interfaces (GUIs) have become the go to method of interacting with computers ever since the Apple Macintosh was introduced in the 1990s (Butow, 2007). The term "Widget" refers to components of a GUI. Over the two decades widgets, like scrollbars, menus and buttons have become standardised. The WIMP acronym has been widely used to describe GUI design, which stands for Windows, Icons, Menus and Pointers (Molina et al 2003). The desktop metaphor leverages real world concepts to provide users with an experience in digital environments where they interact with virtual representations of physical objects like files and folders, on their computer screens.

Three Dimensional User Interfaces (3DUIs) refer to user interfaces that incorporate some form of three-dimensional interaction as discussed in the work of Bowman et al. (2008). These interfaces are mainly considered for their use in MR applications. 3DUIs allow users to interact with objects, environments or information by manipulating them in both physical and virtual spaces (Bowman et al., 2008). These interfaces have gained attention in recent years due to their wide range of applications including interactive visualisation of three dimensional data and their utilisation, within the entertainment industry (Steinicke et al., 2012)

3DUIs provide benefits, such as enhancing the user experience, increasing immersion and improving understanding (Wei et al., 2008). However the usability of 3DUIs in applications generally has not yet reached a sufficient level. There is a need for interaction design solutions that promote natural interaction and effectively utilise the unique features of 3D technologies ("Workshops" 2005). Researchers have explored approaches to designing 3DUIs including the adoption of natural user interfaces (NUIs) that replicate real world interactions while limiting complexities (Arbeláez Estrada & Osorio Gómez 2015). Furthermore there have been efforts to incorporate feedback and proprioception to enhance interactions, with three objects and widgets (Boeck et al. 2006).

In the realm of MR the positioning of displays plays a role in creating an authentic user experience. Display placement significantly impacts the field of view (FOV) and perspective within the environment, influencing how users perceive and interact with MR systems (Bowman et al., 2008). These virtual displays offer real time information or visual cues within the user's field of view without obstructing their view of the environment (Billinghurst & Kato 2002). They serve as a means to present information effectively to users while maintaining a view of their virtual surroundings (Grout et al. 2015).

The choice of virtual display placement should consider factors such as the type of task, the level of immersion desired, and the user's preferences (Barfield et al., 1999). The goal is to replicate the way humans perceive the real world, where objects appear in the correct size and position relative to the user's viewpoint. HMDs users have a small range of focus and so various configuration options for panel interfaces such as flat, curved or surrounded should be considered. Flat tile interfaces tend to feel like a wall and curved tile interfaces track the user's position. Surrounded panel interface configuration curve around the user.

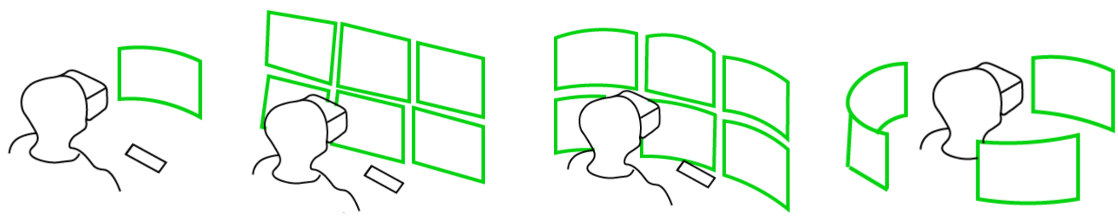

Figure 5. Different display configurations

Head Mounted Displays

Different head mounted displays (HMDs) can vary in terms of their display quality, FOV, tracking capabilities and comfort levels (Ammann Reiffer et al., 2021). For instance some HMDs provide high resolution displays and wide FOV while others may have limitations in these aspects (Blissing et al., 2022). Furthermore the usability, comfort and individual preferences for HMDs can differ among users. It is crucial to consider these factors when selecting the HMD for a specific task (Ammann Reiffer et al., 2021). Ultimately the choice of HMD depends on the application and user requirements.

The rapid advancement of reality makes it challenging to provide recommendations, for headsets or platforms that won't soon become outdated. Therefore the following research primarily focuses on requirements and limitations related to how HMDs interact with the human perceptual system.The goal of HMDs is to replicate and stimulate senses in order to create the illusion of a virtual environment. However, there are limitations in replicating all senses accurately and understanding these limitations can inform software design for a more realistic and effective MR experience.

Hardware Requirements

Virtual Reality can be traced back to the 1960s when head-mounted displays and sensor technology were first developed. It was not until the 1970s that graphical user interfaces, like those seen in today's personal computers, came into existence (Butow, 2007). Since then, the inadequate performance of HMDs contributed to the dominance of 2D paradigms for computer work. For a VR headset to be a legitimate alternative to 2D displays, it must update its image at least as fast as the human perceptual system can process it, and the image must be at a high enough resolution to avoid noticeable artefacts. Until recently, VR headsets were not powerful enough nor cost-effective enough to be a potential alternative to traditional screen displays.

Wang 2023 found that 120fps is an important threshold for VR, as users tend to feel lower simulator sickness symptoms without a significant negative effect on their experience. Chen 2007 found strong support for a threshold of around 15 Hz (equivalent to 90fps) for many tasks, including those that are psychomotor and perceptual in nature. However, other sources push this threshold even lower stating 75 times per second as the minimum update rate for HMDs (Oculus, 2015).

An increase in the frame rate tends to be at odds with an increase in resolution. A higher resolution allows users to see details, textures and small objects with clarity enhancing the overall immersive experience (Peli, 1990). Conversely a lower resolution can lead to pixelation and reduced clarity potentially compromising the experience and making it harder to distinguish visual information (Peli, 1990). Therefore finding the perfect balance, between frame rate and resolution is essential, in creating a comfortable MR experience.

The field of view (FOV), in head mounted displays (HMDs) plays a role in enhancing the user experience. A wider FOV allows for an immersive experience as it expands the user's visual field resulting in a greater sense of presence and engagement (Guo et al., 2018). Research by Polys (2007) has shown that a larger FOV improves user performance in environments with abundant information. To optimise design it is important for designers to take advantage of the FOV and strategically position elements within the user's FOV ensuring they are easily accessible and noticeable (Guo et al., 2018; Spreij et al., 2020). However caution should be exercised when choosing HMD hardware as Sauer (2022) discovered that manufacturers' claims about FOV for consumer HMDs are often unrealistic. Since the human eye has a FOV of around 180 degrees an HMD, with 110 degrees FOV will cover a significant portion of the user's visual field and provide an immersive experience.

The resolution, FOV and frame rate of HMDs all significantly impact the power consumption, another important consideration. Richa et al. (2022) emphasised power optimization in limited power budget systems, like HMDs. Early power consumption estimation at the design stage is crucial for power optimization. The hardware specifications should include power-efficient components and design techniques to minimise power consumption.

Input Methods

A primary concern in interface design is the mode of user input, which is often constrained by the hardware. For instance, smartphones rely on touchscreens, while ATMs use buttons. However, with HMDs, designers have a wide array of options at their disposal for both explicit and implicit style interactions.

MR controllers, motion controllers, and wearable devices offer physical interaction and haptic feedback (Lee & Shin, 2021). On the other hand, VR applications have also employed speech recognition and virtual touch screens as input methods (Win et al., 2022). Most MR applications choose either MR controllers or some combination of hand-tracking or eye-tracking with pose detection, with eye-tracking being available only in the latest headsets. Nevertheless, the MR community remains largely undecided on the appropriate or standard input method to use.

Implicit style interactions enable more intuitive and natural human-computer interactions (HCI) by allowing users to interact with the system through movements of their arms, hands, head, or eyes. This type of interaction can provide a more immersive and seamless experience for users, as it mimics real-world actions and gestures (Jacob, 2001). However, designing and implementing implicit style interactions can be a more complex task compared to explicit style interactions, which involve direct input through buttons or touchscreens.

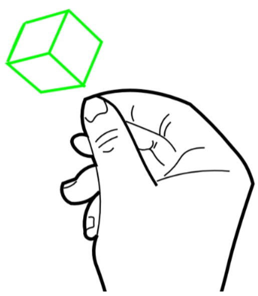

Figure 6. Illustration of Inspection and Manipulation

These diverse input options, each with its advantages and considerations, provide varying levels of immersion, ease of use, and precision, presenting designers with the challenge of finding the best fit for their system's purpose. Designers need to consider the strengths and limitations of each input method when developing MR applications. Factors such as accuracy, latency, ease of use, and user comfort should be taken into account. Additionally, the compatibility of the input method with the specific HMD and software platform should be considered to ensure optimal performance (You et al., 2021).

The choice of input method for MR applications depends on factors such as the nature of the task, user preferences, and the level of realism desired. For example, gesture-based input methods can provide a more immersive and intuitive experience, particularly for tasks that involve physical manipulation or interaction with virtual objects (Wang et al., 2021). Wearable devices and motion controllers offer a more tangible and haptic interaction, allowing users to physically grasp and manipulate objects in the virtual environment (Lee & Shin, 2021)

Methodologies

Designing HMDs involves various methodologies and processes to ensure effective and engaging user experiences. Presence, the subjective experience of being in a virtual environment, is a key factor in VR design (Slater & Wilbur, 1997). Slater and Wilbur proposed a framework for immersive virtual environments (FIVE) that discusses the role of presence in VR. Witmer and Singer developed presence questionnaires to measure the factors underlying presence and found a positive relation between presence and task performance in virtual environments (Witmer & Singer, 1998). Immersion, another important aspect, refers to the extent to which a VR system can deliver an inclusive, extensive, surrounding, and vivid illusion of a virtual environment (Slater & Wilbur, 1997). Pausch et al. demonstrated that users with VR interfaces completed tasks faster than those with stationary monitors, highlighting the benefits of VR immersion (Pausch et al., 1997). The use of ontologies in VR development processes has also been explored, with ontologies being incorporated in various stages of the VR construction lifecycle.

3. VR for Computer Work

In this report the term "Computer work" refers to tasks and activities that involve using a computer to process information, perform tasks, and achieve goals through software applications and digital tools. HMDs are an evolving technology that have the potential in the landscape of computer work. MR applications are gradually making their way into practical and professional applications for productivity enhancement Wexelblat (1993). HMDs have been proposed as an alternative to tackle challenges like privacy, space, lighting, and noise in working environments. The benefit of this realising this potential opportunity became apparent during the COVID-19 pandemic where individuals frequently transitioned from working at the office to at home and vice versa.

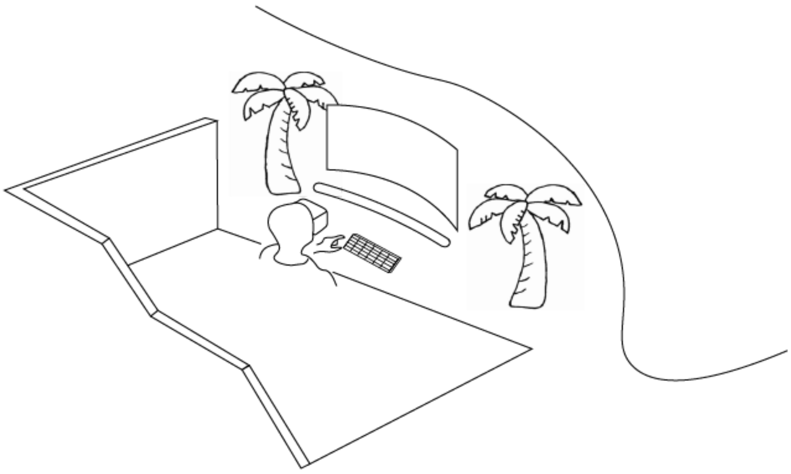

Figure 7. An illustration of a computer work application in a virtual environment

Background: Adapting the 2D Paradigm to VR

MR is fast becoming a new medium to expand the horizons of what's possible with computers. Researchers and developers are looking into different ways to adapt the traditional 2D paradigm of computer work for MR. These efforts span from the visualisation and representation of data to innovative distance communication and educational tools (Weiss, 1998). Pirker (2020) highlights how MR provides innovative forms of interactive learning and working. Hoffmann 2006 proposed a compact, fully immersive VR system that complements the classic desktop workplace, which could be used for applications such as data analysis, design review, and product development.

Beyond this, the concept of spatial computing—a convergence of virtual and physical spaces—has begun to shape the landscape of digital interactions (*Human interface guidelines Apple*, 2023) This is marked by an immersive experience that allows users to visualise and interact with digital information within 3D space, enhancing their understanding and engagement with the virtual content (Billinghurst et al., 2015). In the context of computer work, spatial computing offers new possibilities for data visualisation, collaboration, and problem-solving. It enables users to manipulate and analyse spatial data in a more intuitive and immersive manner (Liao et al., 2019)

These technologies enable the interpretation and understanding of spatial relationships, the processing of large spatial datasets, and the creation of realistic and interactive virtual environments (Ye & Hua, 2013). The design and implementation of spatial computing systems require careful consideration of factors such as presence, immersion, and user experience (Anselin & Rey, 2012). This finding is supported by evidence of a decrease in distractions among individuals working within open office setups when using virtual reality. Users of MR office solutions have self-reported entering a state of flow and expressed a preference towards these alternative MR solutions.

Potential Applications of VR in Computer Work

Numerous studies have explored the potential of HMDs in computer work settings. For instance, Walker (2017) found that they demonstrated utility in aiding users with everyday computer tasks like typing on physical keyboards. Similarly, John (1999) proposed utilising workspace control diagrams and HMDs as viable alternatives to multiple monitors, especially in environments rich with information.

Virtual Displays

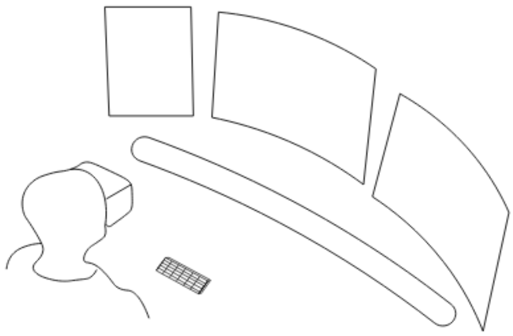

Figure 8. An illustration of virtual monitors

Research findings indicate that virtual displays have the potential to be employed in practical work scenarios, particularly for shorter durations, as virtual displays are not limited by the constraints of physical space and can customise their workspace according to their preferences and needs. Virtual displays also have the potential for increased productivity as users may be able to multitask more efficiently or view information from different sources simultaneously optimising memory utilisation, and physical navigation.

Virtual displays offer the advantage of not occupying any real space and can be easily generated or removed according to user needs. Physical monitors, however, are costly, space-intensive, and lack portability due to their weight and size. Additionally, research has demonstrated that working with both physical and digital documents through AR is possible and well-received by users.

Virtual displays have shown potential feasibility for performing productive tasks; however, technical limitations currently restrict their full potential. HMDs face constraints related to lower brightness and resolution which may hinder the usability of virtual displays. Current advancements suggest that HMD capabilities will improve significantly while becoming more affordable in the coming years with the wider adoption of HMDs.

Pausch et al. (1997) compared the performance of users with MR interfaces to those with stationary monitors and hand-based input devices in a search task. They found that users with MR interfaces completed the task faster. Akpan & Shanker (2018) conducted a meta-analysis of studies comparing the effectiveness of 2D displays versus 3D displays in completing various tasks. They found that the use of 3D visualisation can improve user performance. However, the debate on the need for 3D versus 2D displays continues, highlighting the importance of further research in this area.

The Promise of VR in Computer Work

One of the most compelling promises of MR is the enhanced sense of presence. Traditional computer interfaces, although efficient, often lack the ability to fully engage the user. In contrast, HMDs offer a heightened sense of immersion. Witmer & Singer (1998) found a positive relation between presence and task performance in VR environments. The integration of computer vision and artificial intelligence enables the creation of realistic and interactive virtual environments (Billinghurst & Kato, 2002). Such an immersive experience allows users to feel truly "present" within the virtual space, leading to increased focus, engagement, and potentially improved productivity during computer work (Witmer & Singer, 1998).

Another substantial benefit lies in improved visualisation capabilities. In professions such as architecture, engineering, and scientific research, the ability to visualise simulations, complex data and models in three dimensions can be invaluable. VR offers this capability, providing a unique, immersive visual experience that enhances understanding, analysis, and decision-making (Gong et al., 2020; Win et al., 2018).

MR provides an immersive experience that enhances the understanding and comprehension of complex data by allowing users to interact with the data and explore it from different perspectives. It enables users to have a sense of presence in the data and facilitates the discovery of patterns and insights. VR visualisations can also be collaborative, allowing multiple users to interact and explore the data together (Royston, 2016).

In an increasingly globalised world, the need for effective collaboration tools has never been more crucial. MR introduces a paradigm shift in enhanced collaboration. Through shared virtual spaces, teams, whether they are in adjacent rooms or on different continents, can work together in real-time, breaking down geographical barriers and fostering robust communication and teamwork (Gong et al., 2020; Yoshimura & Borst, 2020). Thoondee (2017) found that MR applications, when designed for relaxation, effectively reduced stress and improved relaxation among office workers.

Personalization in the digital workspace is another area where VR shines. With customizable workspaces, users can tailor their virtual environment to meet individual preferences and needs, from layouts to tools. Such a level of customization can drastically optimise workflows and boost productivity, making tasks more intuitive and efficient (Budziszewski et al., 2016).

Moreover, MR offers invaluable opportunities for simulated training and prototyping. Industries ranging from aviation to healthcare can utilise MR for realistic training scenarios or prototype testing, reducing risks and resource costs (Budziszewski et al., 2016).

Inclusivity, a cornerstone of modern digital ethics, is also addressed by MR. With its potential to offer tailored accessibility features, MR ensures that individuals, even those with disabilities, can fully participate in computer work. This accessibility and inclusivity is a testament to the potential to foster a more inclusive digital future through this new medium (Budziszewski et al., 2016).

MR also reduces physical space requirements by eliminating the necessity for physical workstations, providing much-needed flexibility, particularly in space-constrained environments (Budziszewski et al., 2016). Moreover, MR introduces novel interaction paradigms that extend beyond conventional methods like the mouse and keyboard, enabling users to interact with digital content in more intuitive and natural ways (McMahan et al.,2012).

Input

Text input is an interesting challenge in the context of HMDs. With a sufficiently powerful headset with perfect passthrough capabilities, a keyboard could be used as normal. However, in the absence of such a device, alternative input methods for text entry need to be explored.

Controller-based text input techniques were investigated by (Boletsis & Kongsvik, 2019), who conducted an empirical evaluation of four techniques: raycasting, drum-like keyboard, head-directed input, and split keyboard. The study compared their performance and usability, providing insights into well-established MR text input techniques. Speech recognition can be useful for tasks that involve text input or issuing voice commands (Win et al., 2022).

Gesture-based input methods have also been explored, such as hand-up and hand-down postures, which offer natural and intuitive interaction for tasks like text entry (Wang et al., 2021). Gesture-based input methods were also explored by (Lu et al., 2020), who focused on hands-free text entry techniques in MR, such as hands-free typing and dwell-free typing, discussing their potential for text input in MR systems. Dmitry et al. (2021) addressed the challenge of low-burden responses to open questions in VR. They compared different text input techniques, including controller drumming, freehand typing, and pinch typing, for free-text responses in MR user studies. These methods showed comparable performances and usability, making them suitable for low-burdening text input in MR.

One study by Knierim et al. (2018) explored the use of physical keyboards in VR, tracking the user's hands and visualising them in the virtual environment. The study found that experienced typists benefited from seeing their hands, reaching almost outside-VR performance. Similarly, Yildirim & Osborne (2020) compared 2D and 3D keyboard layouts in VR and discussed implications for text entry tasks with HMDs.

The utilisation of physical keyboards in MR extends beyond their primary function as input devices for typing characters, according to Daniel Schneider and his colleagues. They argue that these keyboards can be tailored to cater specifically to different MR applications and interactions. With the unique attributes inherent in physical keyboards such as multiple keys, haptic feedback which provides tactile sensations upon keypresses, and widespread user familiarity with this traditional input method, physical keyboards offer a familiar and efficient means of text input in virtual environments.

Addressing Challenges

Challenges of MR 3DUI include the learning curve associated with using them (McMahan et al., 2012; Carter & Egliston, 2021). Technical limitations, as outlined by Soares (2021), such as the inferior usability of virtual monitors compared to their physical counterparts, must be addressed. Moreover, ergonomic concerns play a pivotal role. Studies have pointed out discomfort caused by frequent head rotations, the weight of MR headsets, and issues related to visual displays and hand-held input devices (Kim, 2020; Nichols, 1999). Addressing these challenges is crucial to ensure the long-term viability and user-friendliness of MR in computer work settings. As advancements in HMD technology persist, these challenges are progressively being tackled.

Positioned distant from displays, devices like keyboards can cause excessive head rotation, leading to discomfort. Yet, orienting displays cylindrically can reduce head movement. Alternating between virtual and physical displays may induce visual fatigue, but a study by Pavanatto et al. suggests that a hybrid approach might be optimal.

Methodologies in VR Design and Evaluation

The introduction of HMDs in computer work mandates rigorous methodologies and frameworks for its design and evaluation. Various researchers have emphasised the need for structured design evaluation techniques, such as questionnaires and frameworks, to ensure effective MR design (Witmer & Singer, 1998).

A critical aspect of MR is the measurement of presence or the sense of being in a virtual environment. Ways of measuring presence include the presence questionnaire developed by Witmer & Singer (1998). Researchers have also explored the relationship between body movement and presence in immersive environments (Slater & Steed, 2000)

In addition, further comparative performance analysis is important for offering insights into MR's efficiency and comparing performance between MR interfaces and traditional methods. Pausch et al. (1997) found VR interfaces to be quicker in certain tasks.

Researchers and developers have proposed methodologies and frameworks, such as the question-based spatial computing approach, to guide the design and implementation of spatial computing applications (Studer & Hofer, 2019).

4. Creating Comfortable and Sustainable Interfaces

In the realm of MR, the placement of content is crucial in creating immersive and engaging experiences. One of the most attractive aspects of virtual reality is its seemingly limitless potential for content. Unbounded by the limitations of working within two dimensions, users can be teleported to different locations instantly and theoretically place content wherever they want. This seemingly boundless potential is a big part of the draw users feel towards MR.

However, as we redesign computer interfaces, deeper thought needs to go into two areas: where we place content and how we design the user interactions with this content. This is necessary if we want users to engage in deeper use of the technology for longer durations within virtual spaces.

With regards to the first question, we can establish general guidelines for content zones based on an understanding of technical and ergonomic factors that influence depth perception and limit where content is comfortably visible.

Establishing Environment Bounds

As we are attempting to redesign computer work which is primarily done in a seated position, we will assume the user is seated on a non-rotating chair at a desk. Placing the user at the centre of our design environment, we will also assume that they have an average interpupillary distance of 63mm (Pupillary distance PD, 2020) and a field of view of 100 degrees when wearing the headset, as this is the average for modern headsets.

As eyes reduce their focus length they gradually become more and more strained until eventually the eyes are crossed. Although it appears that the eyes are always somewhat strained when looking at close content, Chu et al suggest that eye strain is noticeable at focus distances of between 1 and 0.5 metres and significant at focus distances less than 0.5 metres. Meta suggests that, at the minimum, content should not be placed within 0.75 metres of the user's head. (Alger, 2016)

Expanding from 0.75 metres to 10 metres there appears to be a significant enhancement of the perception of depth and separation between different components. The designed focal distance for the Quest 2 optics, which are by far the most popular headsets currently, is 1.3 metres. However, strong stereo separation tapers off as the distance increases beyond 10 metres. (Alger, 2016)

At a certain focal distance, the user will lose depth perception due to limited headset resolution. The maximum distance of depth perception (MDDP) for a user of any given headset can be estimated as a function of the resolution, Field of view and inter-pupillary distance of the user.

We can calculate the incremental horizontal eye rotation required to move focus from pixel to pixel, based on the resolution. This is significant, as depth perception primarily depends on the angle of eye rotation needed to overlap the images seen by the left and right eye.

$$Required\ incremental\ eye\ rotation\ =\ Field\ of\ View\ /\ (0.05\ *\ Horizontal\ Resolution)$$

Assuming a standard interpupillary distance is 63mm we can calculate the maximum distance the user can perceive depth based on the distance between the eyes and the angle they need to be at to converge on the focus point.

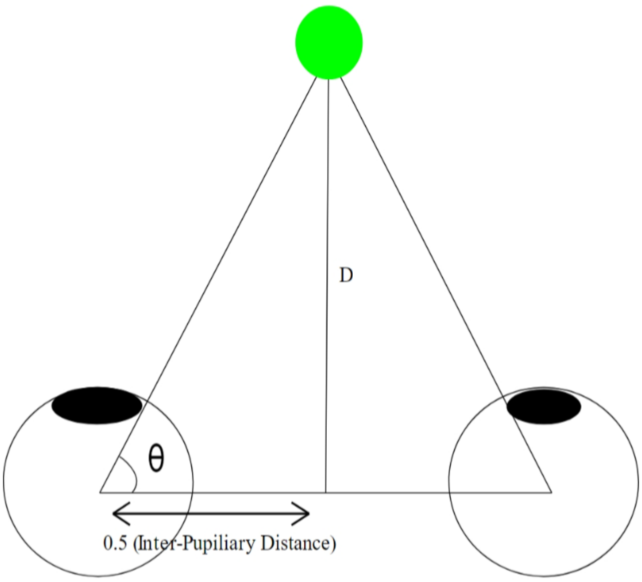

Figure 9. Diagram of ocular convergence and depth perception (Alger, 2016)

$$tan(\theta) = \frac{d}{0.5(IPD)}$$

$$\downarrow$$

$$d = 0.5(IPD) \cdot tan(\theta)$$

Figure 10. a)
Deriving distance as a function of eyes convergence angle and IDP

When the angle of eye convergence gradually reaches 90 degrees, both eyes perceive the same image pixel for pixel. This results in an essentially monoscopic visual experience. To find the MDDP we simply subtract the incremental horizontal eye rotation from that 90 degrees to estimate the maximum distance for perceiving depth. This is the farthest focal point where the eyes are not seeing the same image pixel for pixel.

$$d = tan\left[90° - \left(\frac{FOV}{.5R}\right)\right] \cdot \frac{IPD}{2}$$

d = maximum perceived distance
FOV = device horizontal field of view
R = device screen horizontal resolution
IPD = user inter-pupillary distance

Figure 10 b) Equation for an estimate of MDDP (Alger, 2016)

Here is a table of the estimated MDDP of notable headsets. What's evident is that the MDDP varies significantly, meaning that there are no stable outer bounds for where 3D content can be meaningfully placed and designers will have to assess the specific limitations and capabilities of each headset when determining the placement of content in HMDs.

Meta Quest 2	FOV 97°	Resolution: 1832p	MDDP ~ 17.0m
Pico	FOV 105°	Resolution: 2160p	MDDP ~ 18.6m
Apple Vision Pro	FOV: 110°*	Resolution: 3680p	MDDP ~ 30m

A headset typically has a standard distance between pixels at the centre of the FOV where the eyes focus most effectively. The same minute adjustment to the rotation of the eye is needed to move from one pixel to the next. This means that the higher the resolution of the display the smaller the eye rotation increments will need to be to move between pixels. This has a significant impact on depth perception as if the resolution is too low, the user may observe a pixelation effect or "screen-door effect", causing a decrease in the perceived depth and realism of the virtual environment.

Establishing Comfort Zones

People can comfortably rotate their heads at 30° and maximally rotate their heads at 55°. Rational estimates for comfortable and peripheral vision can be achieved by combining head rotation with half the FOV. Content placed from 80° either side of the straight-ahead position can be comfortably seen by the user and content placed from 80° to 105° can be seen in the peripheral vision. Likewise, people are comfortable rotating their heads up 20° and down 12° and can maximally rotate their heads up 60° and down 40°. These parameters give us a frustum of comfortable and peripheral vision that can be utilised for placing content in VR. (XR Design Theory and Practise for Digital Eyewear, 2020)

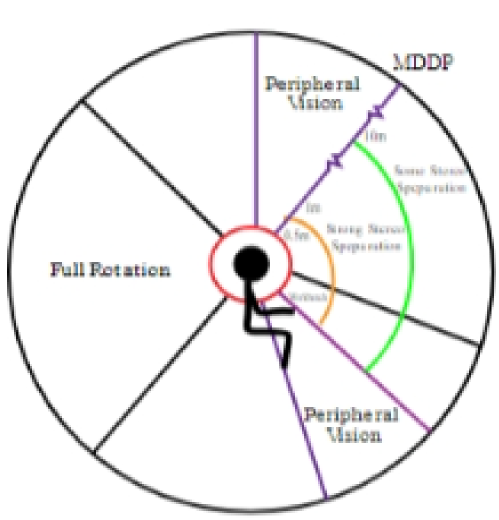

Figure 11. Side view of proposed content zones

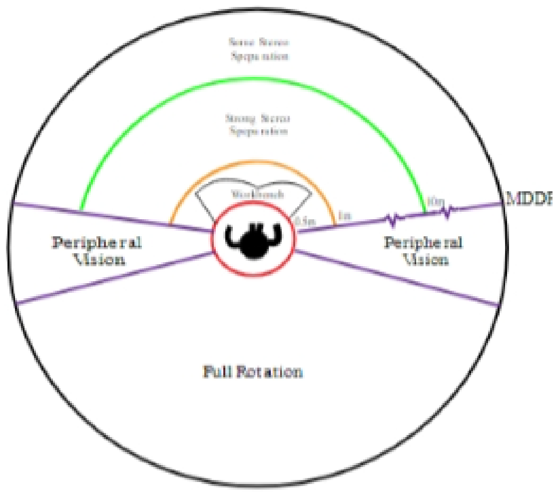

Figure 12. Top down view of proposed content zones

Now that we have a clear understanding of where we can place virtual content we can approach the subject of designing interactions with them. We can narrow down a zone in which these interactions can take place by once again looking towards anatomy. The effort expended by the user to interact with the system should be minimal and therefore arm extension should be limited to 2/3rds maximum arm extension. This is the range in which desktop elements such as keyboards and mice are easily manipulable without stretching. Again, we must remove any area within 0.5 metres of the user, as any content placed here would be uncomfortable to view. The remaining space we will call the "workbench zone" where users can move their hands to select a tool to manipulate objects or type comfortably for prolonged periods.

Proxemics

In the context of MR, proxemics can guide us in designing virtual environments that resonate with our natural instincts and comfort levels. As elucidated by Witmer & Singer (1998), presence, the feeling of "being" in a virtual environment, is of paramount importance. To elevate this sense of presence, content — particularly virtual displays and 3D objects — must be positioned in alignment with our natural understanding of space.

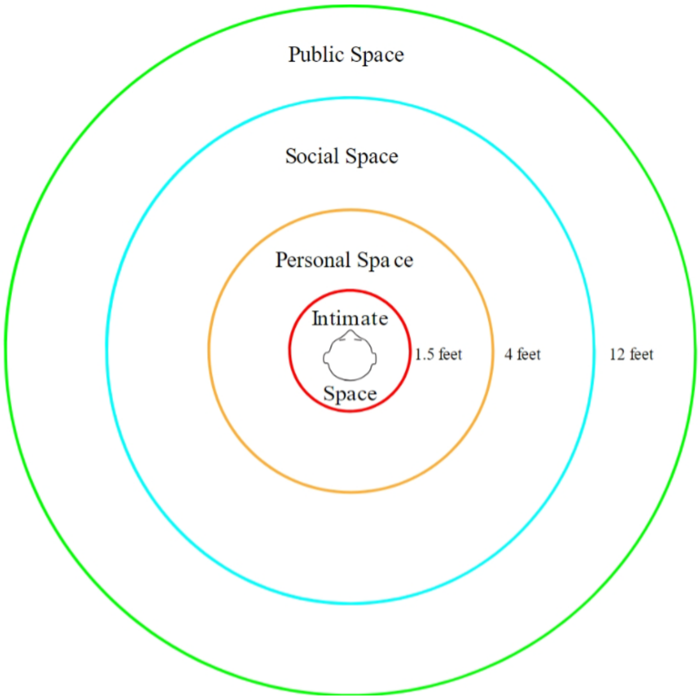

Figure 13. Edward T. Hall's The Interpersonal Framework

At its core, proxemics is the study of human spatial relationships, or how we perceive our personal space during social interactions. Personal space can be fluid, however, Edward T. Hall's classification provides a clear delineation:

1. **Intimate Distance (0 to 18 inches):** This zone is reserved for our closest relationships. Physical contact is frequent here. Translated to MR, elements placed within this zone must be of utmost importance.

2. **Personal Distance (1.5 to 4 feet):** This zone is for interactions with friends and family. In MR, content placed here could be for frequent, casual interactions.
3. **Social Distance (4 to 12 feet):** This is the zone of formal interactions. In MR, it can be a space for elements that users need to be aware of but don't interact with constantly.
4. **Public Distance (beyond 12 feet):** Typically for addressing a wider audience. In MR, this zone can contain elements of the broader virtual environment or backdrop.

By respecting the zones of interpersonal distance, MR designers can determine the best placement for various elements. Critical interactive tools might sit within the personal zone, while ambient information might be situated further out in the social or public zones. This deliberate positioning impacts user comfort, detectability of content, and consequently, engagement.

5. Human-Centric Design Guidelines

Redesigning an application for MR can be a daunting task as it requires a thorough understanding of the capabilities and limitations of the technology as well as the needs and preferences of users. Designers of MR interfaces face significant challenges in conceptualising and methodologically approaching the overall design process (Tanriverdi & Jacob, n.d). With the richness and complexity of objects, behaviours, interactions, and communications in MR interfaces, designers must think comprehensively about the overall design while also breaking the design task down into smaller, conceptually distinct, and manageable tasks. Communicating the structure of the design to software developers is also a challenge, as virtual reality interfaces involve a combination of 2D and 3D elements that need to be integrated seamlessly.

Instinct-Driven

Human beings are “hard coded” to respond to external stimuli received through the senses. Software designers can leverage these human instincts, natural tendencies and innate behaviours to create intuitive user interfaces that elicit an involuntary user reaction through means such as typography, colour palettes, and composition techniques.

By understanding and capitalising on these inherent tendencies, designers can employ visual elements that intuitively guide user behaviour without the need for explicit instruction or training. This approach taps into our innate responses to certain stimuli based on evolutionary factors gained throughout our history as humans. Through careful consideration of design choices like font style, use of colour theory, and thoughtful arrangement of graphical components within interfaces, designers have the power to create seamless experiences that align with human instinctive behaviours.

When it comes to virtual reality headsets, the interaction with the human perceptual system is significantly heightened compared to traditional screens. This emphasises the need for careful consideration of these instinctive or automatic responses that users may have. MR headsets can effectively substitute 8 of the 10 variables found in the plenoptic function, which is responsible for describing a user's vision. This substitution results in users perceiving and accepting the presented version of reality more readily, both consciously and subconsciously. As a result, we now possess the ability to enhance and facilitate more comprehensive forms of human-computer interaction than what was previously possible through traditional 2D interfaces.

Regrettably, inadequate design can result in negative consequences such as inducing feelings of discomfort and nausea. For instance, this may occur when there is a mismatch between the movement of the virtual experience's visual perspective and the user's actual head movements. It is, therefore, crucial that the design of virtual reality applications is carefully developed and perfected to provide an optimal user experience, especially when considering potential extended usage throughout the workday.

Three-Part Framework

The foundation of this research lies in the assertion that 2D applications are not only still relevant in the world of MR but are, in fact, indispensable. They offer users a sense of familiarity, drawing upon ingrained mental models of interface interaction. Yet, it's also acknowledged that a mere transplant from 2D to MR isn't enough. The design needs to be refined and enhanced, making the most of MRs unique features, such as depth perception, panoramic views, and spatial cognition. Upon extensive research and analysis, three pivotal interface elements were discerned for designers embarking on this transition:

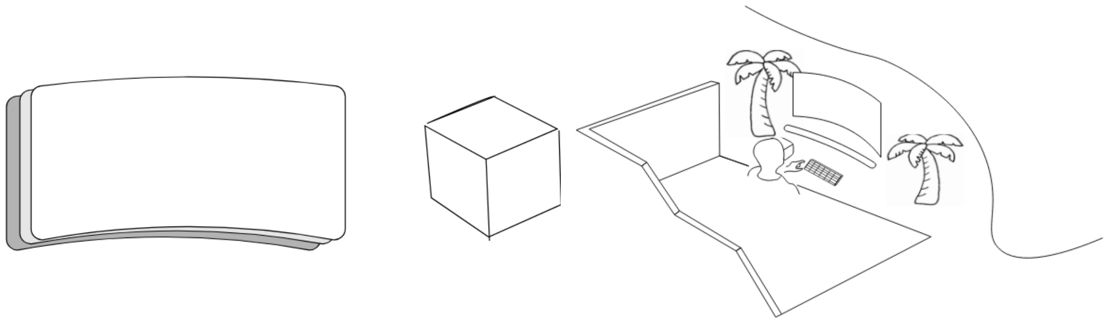

Figure 14. Diagram of windows, volumes and virtual environments, from left to right

1) **Virtual Displays/Windows:** There exist many potential benefits to redesigning computer tools to fully take advantage of HMDs; the most immediate and compelling is the potential for 360-degree monitor displays. Instead of being confined to the dimensions of a 2D screen, users could have a panoramic workspace that extends around them in all directions. This would allow for a more immersive and visually stimulating work experience, potentially increasing focus and productivity. Additionally, users would have to spend less time and energy managing the abstraction between tasks inherent to the current 2D window system allowing for more of their attention being able to be committed to the productive work itself.

2) **Volumetric Interfaces/3D models/Volumes:** Users would also benefit from the potential for entirely new volumetric interfaces, ones specifically designed for the volume inherent to HMDs. For example, users could manipulate and interact with virtual objects in 3D space using gestures, allowing for more natural and intuitive interaction. Volumetric interfaces could range from a simple battery indicator with volume to a complex 3D data visualisation, from a scan of a real-world object to the model of one being designed. By redesigning computer work for HMDs, we can create interfaces that take advantage of the spatial capabilities of these devices. This is a particularly interesting opportunity as conventions for such interactions are yet to be fully established, allowing for innovation and exploration in interface design.

3) **Virtual Environments:** Furthermore, an HMD, depending on the degree to which it blocks outside light can alter or entirely replace environmental conditions which are critical to entering a state of flow and enhancing productivity. For example, HMDs could provide virtual workspaces where distractions are minimised, allowing users to focus solely on their tasks. HMDs opens up the

possibility for environment optimisation and customisation, much like in the manner that one might create a dedicated home office space to eliminate distractions or create a clean customised desktop on their PC.

The following 20 guidelines are researched backed assertions about how to approach the design of the above UI components. These are provided with the intent of providing a solid basis for the redesign of 2D applications for HMDs.

User-Centric Design Guidelines

Discovered Design Guideline 1 - Virtual Monitor Placement: Office ergonomics recommends that screens should be placed between 15 and 50 degrees below straight ahead for comfortable viewing. Monitors should be placed in the designed focused distance where the image is sharp and crisp which is particularly important for reading text. This is typically between 1.3 and 2 metres in front of the user. The exact placement of virtual displays will vary depending on the task the user is performing, their personal preferences, and comfort.

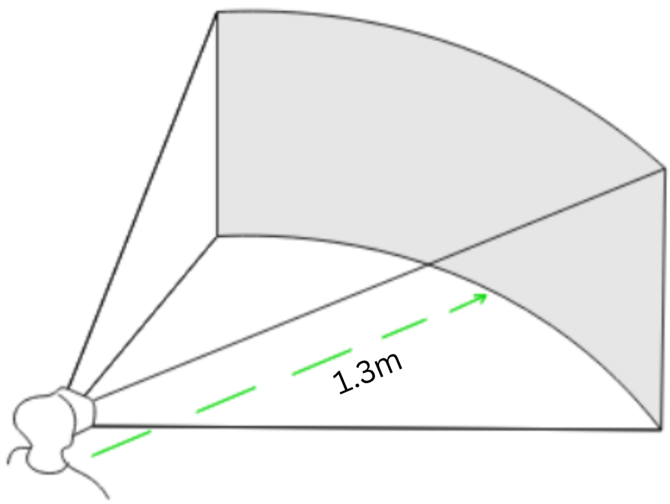

Figure 15. Illustration of optimal distance from virtual monitor

Discovered Design Guideline 2 - Anchor Windows to Scene: Anchoring windows directly to a user's field of view can be disorienting and uncomfortable. Instead, designers should anchor items within the virtual space and have the windows orientation track the user's position, allowing users to have control over their field of view and maintaining a sense of immersion.

Discovered Design Guideline 3 - Use Depth to Establish Hierarchy: Using depth for establishing hierarchy can be invaluable. Nearby and smaller controls can dominate visually over distant content. Subtle shadowing effects between window elements can offer essential visual cues regarding the hierarchy of content. To enhance the sense of depth, designers should employ real-time dynamic lighting techniques, like spotlights or point lights, and avoid static lights with pre-baked shadows.

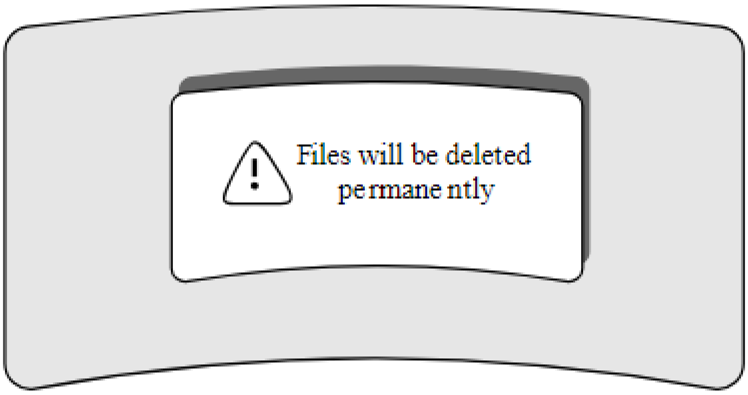

Figure 16. Illustration of windows with depth hierarchy

Discovered Design Guideline 4 - Focus on Clarity and Legibility: Text is one interface element that does not benefit from depth. 3D text, especially when viewed at an angle, can be hard to read. Text within interfaces should remain flat to facilitate clarity and legibility. Font weights and styles should be adapted to emphasise readability. Additionally, Text should not be placed on transparent surfaces as the user's eyes will have to diverge on the background to converge on the text and vice versa resulting in blurry vision.

Discovered Design Guideline 5 - Panels and Pointers: Pointers also allow users to interact with virtual objects or UI elements (Alexandrovsky et al., 2020; Moraes et al., 2021). Allow users to interact with virtual displays dynamically, such as expanding and repositioning them. Users should have the ability to leverage their spatial understanding and memory by positioning elements of the display in their preferred locations. Virtual touch screens provide a familiar interaction paradigm, allowing users to interact with virtual objects using their hands or fingers (Win et al., 2022). For doing this at a distance, the use of pointers has shown advantages in terms of speed, accuracy, cognitive demand, and performance (Alexandrovsky et al., 2020).

Discovered Design Guideline 6 - Embracing Horizontal Ergonomics: According to Witmer & Singer (1998), the user interface, including the size and location of virtual buttons, menus, and other interactive features, can have profound effects on user interaction. Objects meant to garner immediate attention or interaction should be within the user's immediate field of view, ensuring optimal detectability. Those that are of secondary importance can be positioned slightly farther but should still respect the defined interpersonal distances. Reflecting on the natural human range of motion – for many, moving eyes and head horizontally is more comfortable than vertical shifts. To mitigate fatigue, especially for the eyes and neck, position content so it falls naturally within users' line of sight. Wide canvases are preferable to tall ones; for instance, a freeform canvas might stretch more to the sides than upwards. Essential content should ideally be centred for easy accessibility and visibility.

Discovered Design Design Guideline 7 - Address the Tactile Feedback Gap: The absence of physical feedback during MR interactions creates a sensory void. To enhance the interaction, incorporate feedback mechanisms like visual cues, auditory signals, and state changes. For instance, implement button visuals that elevate layers for a tactile illusion. Also consider implementing hover states, and proximity highlights, with accompanying spatial sounds for a comprehensive interactive experience. Designers should take advantage of tactile feedback. For example, virtual poke buttons strategically placed to appear in line with a physical desk can give the user satisfying tactile feedback on selection.

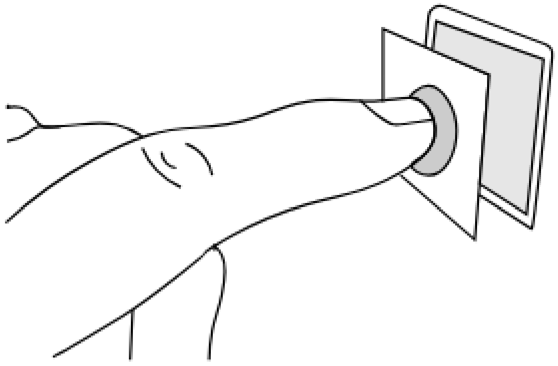

Figure 17. Illustration of a virtual button being pressed

Discovered Design Design Guideline 8 - Customise Keyboard Input: The prevalent interaction tools in traditional 2D computing have been the keyboard and mouse. The tactile sensation of pressing a keyboard key remains significant for users. Additionally, keyboards can function in a variety of ways in VR, and designers should be open to customising how they design interactions with traditional tools.

Discovered Design Design Guideline 9 - Consider User Fatigue: Continuously holding hands aloft can lead to user weariness. Therefore, it's crucial to explore and integrate alternative input techniques like voice commands, trackpads, or keyboards for longer tasks or activities. Eye-tracking technology offers precise control and can be used as a highly accurate alternative input method for users to control distant interface components and reduce hand fatigue.

Discovered Design Design Guideline 10 - Use Familiar Gestures and Feedback: Consistency is vital. Thus, when crafting MR experiences, it's essential to adopt recognized gesture patterns and ensure the UI's responses align with users' prior experiences so that users can focus more on the experience rather than relearning interaction methods. Familiar gestures, such as pinching for scrolling or using both hands for zooming and rotation, should be incorporated. The success of these interactions heavily relies on the UI's feedback mirroring the hand's motion, which ensures users feel a connection to their actions.

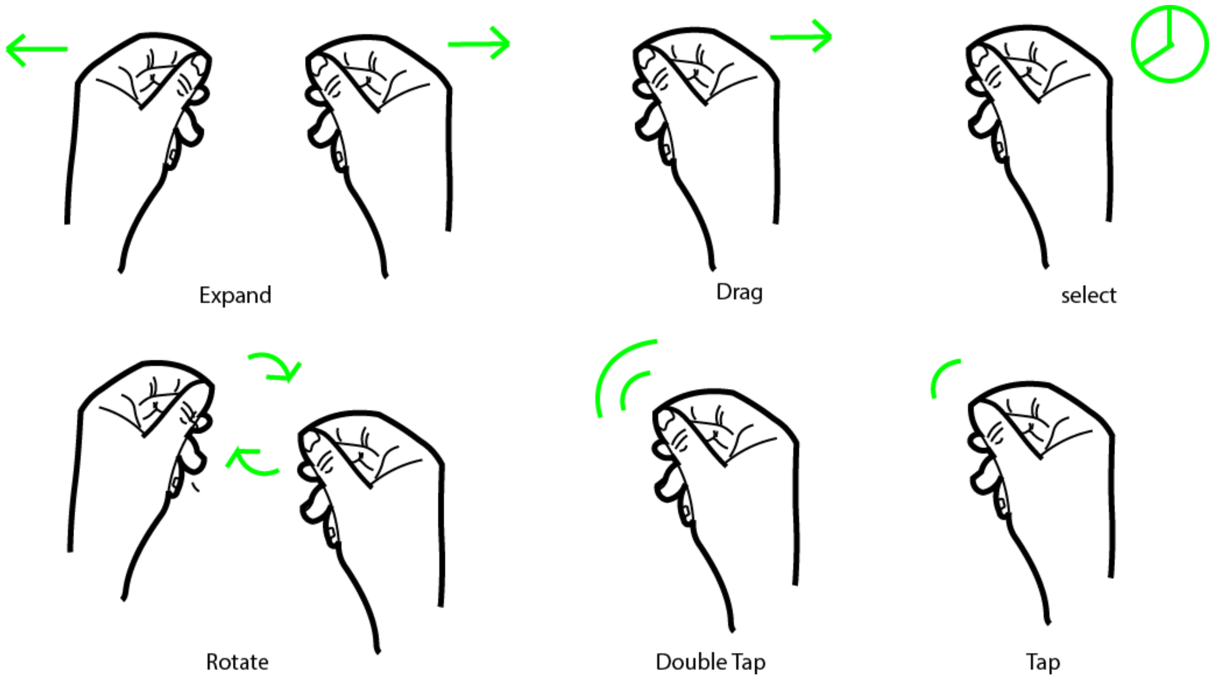

Figure 18. Illustration of common or default UI gestures, similar to (*Human interface guidelines Apple*, 2023)

Discovered Design Design Guideline 11 - Exercise caution with Custom Gestures: Creating unique gestures can personalise an experience, but they should be straightforward to understand and execute and sufficiently unique to avoid clashing with ordinary hand movements. They should be easy to repeat without causing undue strain, have very low chances of accidental activation, and be designed to foster inclusivity for those using assistive technologies. Furthermore, unintended communication or misunderstanding should be avoided by accounting for the cultural and personal significance of gestures.

Discovered Design Guideline 12 - 3D Model Integration: When adapting designs for HMDs, aspects of the application that benefits most from spatial or immersive elements should be highlighted. While many interface elements might remain similar to 2D designs, 3D models can greatly enhance user experience in certain areas. By emphasising these unique spatial moments, designers can create memorable interactions using the tools of depth and scale.

Discovered Design Guideline 13 - Balance Scale and distance: Seinfeld et al. (2022) emphasised the sensitivity of human sensory processing to stimuli proximity. Hierarchy and focus can be determined by using the tool of depth. Nearer objects, due to the ease of inspection from multiple angles, can be smaller whereas content placed further away can be larger, prompting interaction from a distance. Scale serves as a tool for emphasis; smaller objects can feel personal, while larger ones can feel grand. Objects often work well when represented at real-life scale. Designs should remain adaptable, given the variability of users' physical spaces.

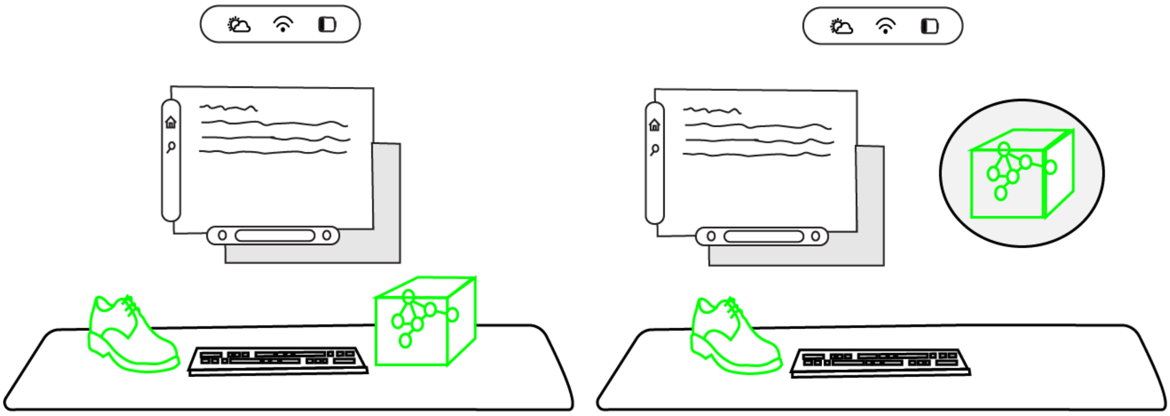

Figure 19. Illustration of different positions for 3D content.

Discovered Design Guideline 14 - Keep environments smooth and simple: Environments should be kept simple, overwhelming users with content can lead to confusion. User immersion in an environment can be enhanced by the introduction of subtle motions, such as moving clouds or rippling water. Additionally, utilising spatial audio and binaural audio that corresponds with visual elements further immerses users. Transitions within the virtual space should be smooth and predictable, ensuring the user feels in control. Users should also have an intuitive mechanism for exiting the experience immediately if desired.

Discovered Design Guideline 15 - Use Translucency to Ground Interfaces: Designing virtual displays with an element of translucency can help them fit into any spatial context and lighting. Avoid heavy solid colours, particularly on windows, as this can result in a more confined and weighty interface feel.

Discovered Design Guideline 16 - Use Content Zones as the foundation of the Redesign: MR design's intricacies demand a deep understanding of human spatial relationships particularly when determining the placement of virtual displays and 3D objects. Through the lens of proxemics and guided by the comprehensive research provided, designers can craft human-centric MR experiences that are immersive, intuitive, comfortable and naturally resonant with users' inherent spatial instincts.

Discovered Design Guideline 17 - Design environments Strategically: Virtual Environments can be used to coax the user towards desired behaviour by leveraging innate responses such as Cathedral Effect, Prospect Refuge Theory, and Attention Restoration Theory. Likewise, strategic use of tools such as Colour, Contrast, Lighting, Audio, Motion and Composition can also influence user behaviour and guide their attention within the virtual environment.

Discovered Design Guideline 18 - Design with Accessibility & Privacy in Mind: Designing virtual reality interfaces requires careful consideration of accessibility and privacy factors. Designs should account for Impairments such as colour blindness or visual impairments by providing alternative colour options or text-reading capabilities. To enhance privacy, a keyboard overlay with shuffled keys can be used which will protect from over-the-shoulder viewing.

Discovered Design Guideline 19 - Promote Presence and Interaction with Content: Presence significantly impacts user engagement and task performance in MR, as noted by Witmer & Singer (1998). Design should align with users' natural spatial perceptions to promote this sense of presence. Objects or displays placed too closely might seem invasive, mirroring real-world discomfort when our personal space is intruded upon. However, positioning them too distantly could reduce user engagement, creating a sense of detachment, as also highlighted by Witmer & Singer (1998) The personal distance allows for comfortable access.

Discovered Design Guideline 20 - Embodiment & 3D Avatars: Embodying a user within an avatar or with 3D virtual objects is a nuanced task. As suggested by Li et al. (2019), the positioning of a virtual representation in relation to the user's real body can influence the sense of ownership and agency. The feeling of having a virtual body or avatar, known as embodiment, gets enhanced when virtual representation aligns with the user's physical self. A misaligned avatar can disrupt the immersive experience. If a user's virtual hand reaches out to touch a 3D object, the spatial relationship should be seamless, ensuring that the perceived distance and interaction feel genuine, minimising simulator sickness. Positioning elements in harmony with natural proxemic behaviours can reduce disorientation, enhancing the overall experience.

Research Conclusion

Thus far an attempt has been made to discover guidelines on how to redesign the 2D paradigm of computer work for HMDs. Previous discussions considered ergonomic implications the redesign of work applications for HMDs so that they are comfortable for the user to wear for long durations. This ergonomic analysis when combined with the suggestion of a 3 component approach and 20 guidelines for the redesign of these components should inform designers about how to redesign 2D computer work applications for use in a virtual environment. The main assumption for the guidelines and methodology outlined in this project is that the user will be seated at a desk with a keyboard in front of them.

The underlying theory behind this framework is that 2D applications are still useful in MR as they provide the user with a familiar interface and leverage preexisting mental models about how to interact with the interfaces. Another benefit of this approach is that designers and developers will not have to start from scratch when redesigning displays. With minor tweaks to the flat 2D interface that give it a sense of depth and an information hierarchy as well as other small tweaks to adapt for the new position of the interface, it is ready for use in a virtual environment.

The following section will describe a demo application designed and implemented to showcase the potential benefits of adapting computer work for HMDs as discussed earlier, and suggest a methodology for future endeavours to consider which may expedite the redesign process.

5. Implementation

Introduction

This project seeks to unravel the nuances of how HMDs can be harnessed to enhance productivity. In earlier sections how to redesign work applications for HMDs based on research is discussed. This section will apply these research findings and in doing so, hopefully contribute foundational knowledge for future endeavours aimed at creating VR tools tailored for computer work.

Goals:

- Demonstrate that 2D computer work applications can be redesigned into useful 3D experiences by focussing on breaking the existing application into smaller components and rethinking their spatial organisation in a 3D environment.
- Demonstrate the potential of each of the three components: 1) Windows, 2) Volumetric Interfaces and 3) Environments for enhancing productivity in computer work applications
- Demonstrate the utility that volumetric interfaces and panoramic displays can have for enhancing productivity in computer work applications.
- Redesign a common computer work task, regularly done in a 2D setting to work in a virtual environment with a volumetric interface.
- Develop an application that demonstrates the potential for improved productivity capitalising on better navigation and contextual understanding.
- Explore the use of Keyboards in a virtual environment and assess their effectiveness when combined with implicit input techniques such as hand tracking.
- Explore the effect of proxemics and ergonomics on the design of an immersive application.

After considering the aforementioned goals of the project and conducting an analysis of productive tasks commonly performed in a 2D computer work environment such as multitasking, data analysis, and document editing, it was evident that the redesign of a research tool that leverages the spatial capabilities of HMDs would be a suitable candidate for this project.

Background and Motivation

Note-taking applications have been a consistent cornerstone of digital tools. Despite their ubiquity, one can argue that they have not seen much innovation. The transition from traditional pen-and-paper journals to digital platforms solved some inherent limitations of pen and paper—space constraints, searchability, and permanence. Yet, digital migration has only scratched the surface of what's possible, especially considering the limitations of linear, folder-based organisational structures which may not mirror the associative, interconnected nature of human cognition.

Human thought processes rarely operate in isolated silos. Ideas mesh and intertwine over time, leading to new insights. Current note-taking paradigms, with their focus on file-based categorization, often don't foster this kind of interconnected exploration. Knowledge is not compartmentalised into folders within our minds but exists in a web of concepts. So, why shouldn't our note-taking tools reflect this innate aspect of human cognition?

The "Detective's Crazy Wall," is an old paradigm that encapsulates this idea of interconnected information. Frequently featured in cinema, it visualises information and its connections, providing a contextual overview of information. Yet, its real-world application is rare, given space constraints.

Digital knowledge graphs combine the expansive visualisation of a detective's wall with the advantages of digital storage. Obsidian.md is a prime example, it constructs 2D knowledge graphs by interlinking markdown files, serving as a tool for researchers to thread together distinct ideas. The knowledge graph paradigm promotes a holistic view of one's notes, facilitating easier pattern recognition and interconnection, much like the aforementioned detective's wall.

However, 2D representations have their shortcomings, especially as the complexity and volume of information grow. Overlapping edges can obscure connections and nuances. It's here that the possibilities of 3D visualisation in virtual reality environments come into play.

Envision a platform that leverages the immersive spatial dimensions of VR to organise notes. Rather than relying on traditional folder systems, notes are categorised based on content and context, represented as nodes within a force-directed knowledge graph. The interlinking of nodes would ensure related ideas gravitate towards one another, providing a dynamic, spatial representation of accumulated knowledge. Such visualisation can adapt and grow organically, mirroring the fluid evolution of understanding over time.

Linking markdown files via the Wikilinks syntax and visualising them as a network or knowledge graph resonates deeply with principles underlying the semantic web, linked data, and knowledge graphs. Conceived by Sir Tim Berners-Lee, the semantic web aims to structure web data for easier machine processing. Similarly, Wikilinks provide structure and interconnectedness to content, akin to the concept of linked data, which promotes structured and interwoven datasets. This interconnected visualisation essentially forms a basic knowledge graph, where nodes (markdown files) symbolise entities and Wikilinks embody their relationships, all of which align with the overarching vision of a more organised and comprehensible digital information ecosystem.

The Redesigning Process for A Notes Application for VR

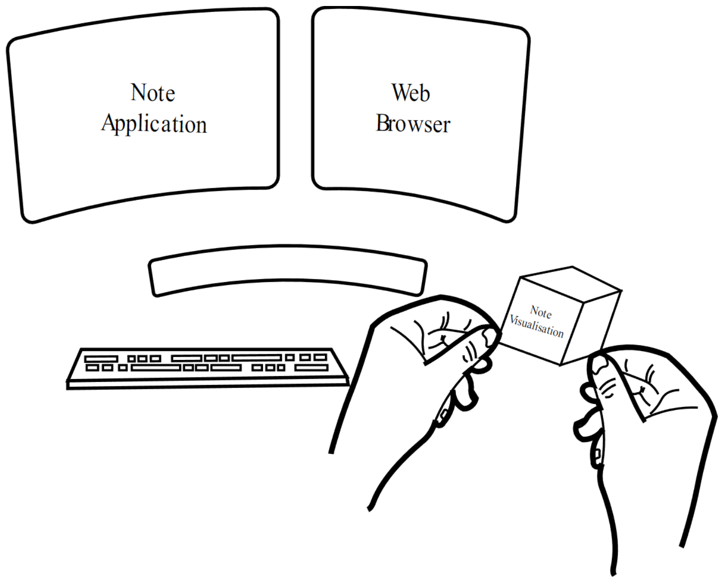

Figure 20. Illustration of concept behind redesigned note application

This project essentially aims to bring a notes application like obsidian into VR by displaying it in a web browser tab floating in 3D space. The user will be able to interact with this notes application using their keyboard which is tracked in physical space and displayed in the headset so that they can see the keys that they are typing. Beside the window with the notes application running in it there will be a 3D visualisation of the notes in the notes window, as represented by a 3D knowledge graph visualisation where every note is a node and every relationship between notes is a link. Relationships between notes are signified using the wiki links syntax. The user can manipulate, grab and scale this knowledge graph.

Step 1: Conceptualization and Requirement Gathering

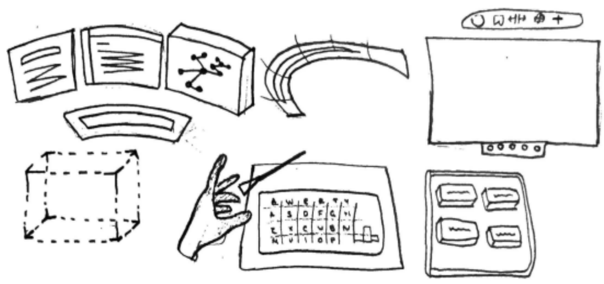

Figure 21. Initial Sketches

Figure 22. User Persona

Purpose and Core Functionality: The process was started by questioning the purpose and core functionality of the application. Questions such as "”why redesign the application? and "how will it benefit from transitioning to a VR environment?" were asked. The primary goal of the notes application was to provide users a platform to capture and organise their thoughts efficiently. (See section background and motivation)

Sketches: Visual representation helps in understanding the spatial needs and layout of the new design.Initial sketches were developed to envision the transition of traditional note-taking elements to a 3D spatial setting. These sketches offered a blueprint of how notes, links, and related functionalities might appear in a VR space.

Scenarios: A series of user scenarios were crafted, outlining hypothetical interactions with the application. These scenarios aided in identifying potential challenges users could encounter when accessing and navigating their notes in this new format. (See section Scenarios abstract)

Task Analysis: Each function of the notes application was dissected to its basic elements, ensuring that essential features were retained in the VR redesign. This exercise was instrumental in streamlining the application, making it more apt for a VR setting.(See section Task Analysis in abstract)

Technical Feasibility Analysis: An in-depth assessment was undertaken to gauge the technological capabilities available. This preemptive analysis ensured that the design envisioned was not only desirable but also achievable.(See section Technical Feasibility Analysis in abstract)

Step 2: Structuring and Positioning

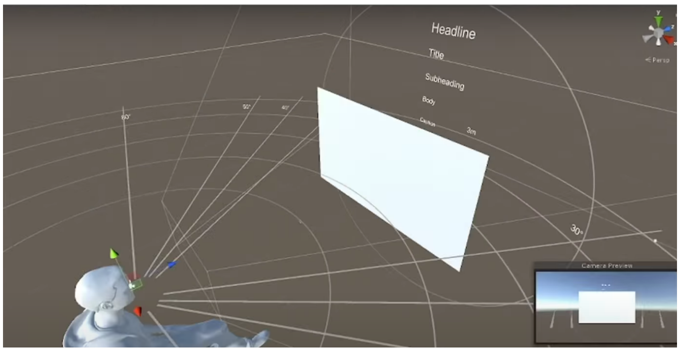

Figure 23. Ergonomics visualised in Unity Engine

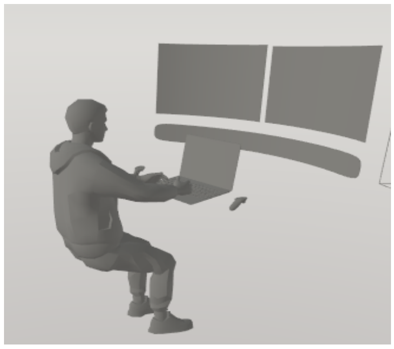

Figure 24. Greybox of concept

Greyboxing: A basic layout was devised using greyboxing, which visually conceptualised the initial layout and spatial positioning of interface elements. Using simple shapes and forms to represent elements allowed for the exploration of how components like windows, and menus might exist and interact in a virtual environment.

Virtual Displays/Windows: The virtual display area was broken up into different sections based on proxemics so that the user would have a more natural understanding of the space, with personal content such as the notes apps close to the user and public content such as web browsers pushed back into the public space. Intuitive hand controls and gestures for navigating and manipulating these displays were also incorporated.

Volumetric Interfaces: Key areas for 3D models or volumetric data representations were determined. This includes data visualisation, UI indicators, or any interactive spatial component. A significant challenge here was determining the placement of the knowledge graph visualisation to ensure optimum interactivity, user convenience and engagement.

Information Hierarchy: Depth was leveraged as a tool to categorise information and create a hierarchy based on importance of information. Consistently used features, like toolbars, were positioned nearer to the user, in a separate window, indicating their prominence and frequent use.

Step 3: Prototyping and Testing

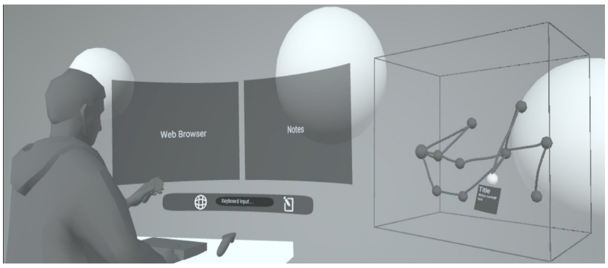

Figure 25. Adding more detail to design

Figure 26. Building Prototype from greyboxes

Prototyping: The grey boxes from the previous step were converted into models with textures and then into prototypes with limited functionality. These were not highly detailed however they allowed for navigation and interaction, offering a semblance of the final product. The placement and size of interface elements were established and their usability and effectiveness was tested in a 3D environment.

Test the application from the inside: Real-time interactions with users were simulated, where the application was presented as fully functional however the application was manipulated by designers testing the interaction styles. This was conducted using shapes XR which allows for collaborative manipulation of virtual environments and scenes. Emphasis was placed on assessing user comfort and overall usability.

Step 4: Iteration and Refinement

The design underwent several iterations, each time refining features enhancing the overall user experience. Issues were addressed and interactions enhanced with more advanced prototypes until the design feels mature and ready for development.

Design of Redesigned Note Application With Volumetric Graph Interface.

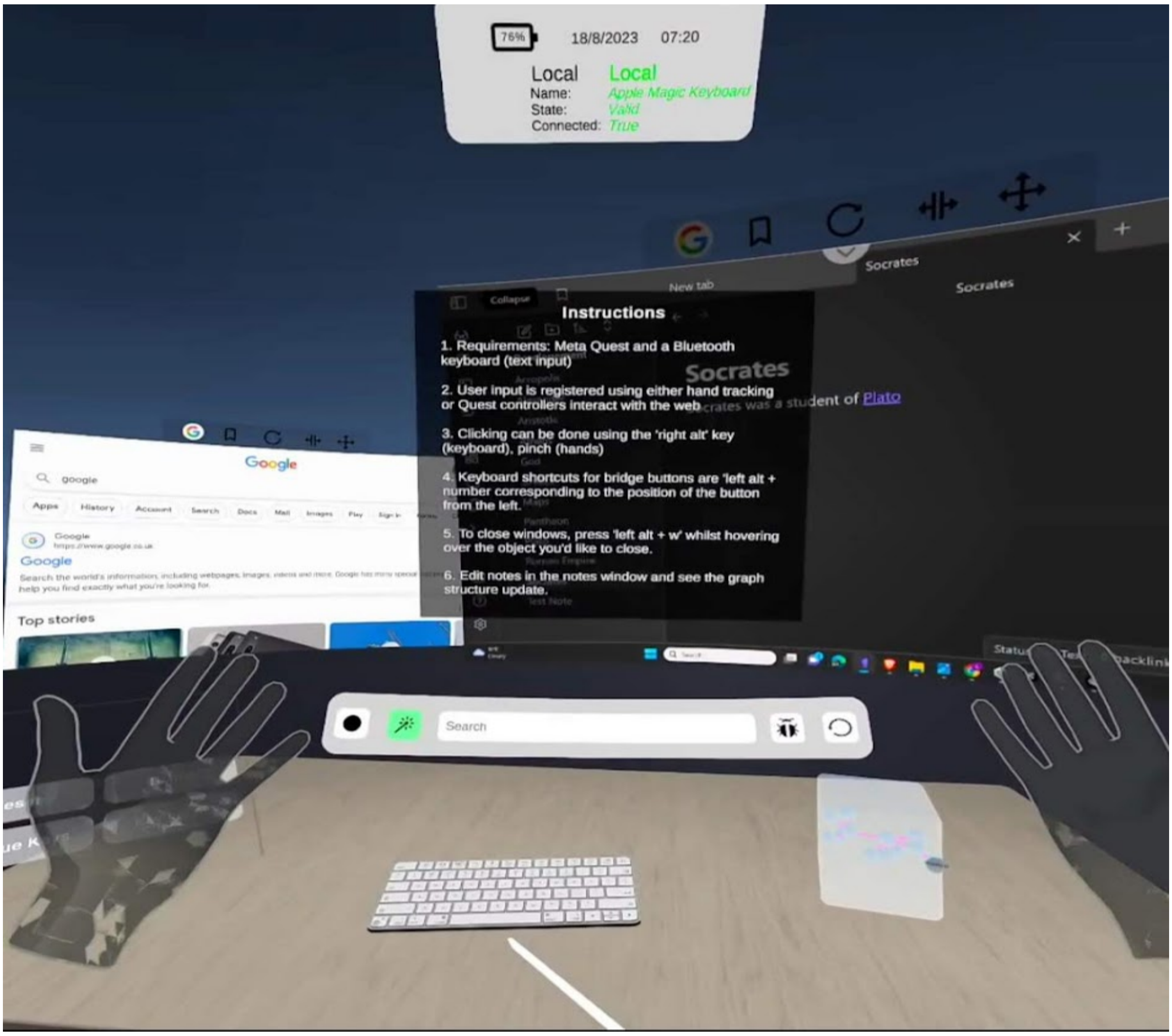

Figure 27. A screenshot from the developed application

An interaction with the system might go like this. A user is in Vr with a notes application in front of them. They are sitting at their desk in the real world and in VR a desk is displayed in front of them to mimic this. On the virtual desk is a digital representation of their actual physical keyboard which is being tracked in 3D space. This way they can see the position of their hands and keyboard and therefore type.

In Front of the user floats a notes application. They can select this notes application using a hand pointer to select this tab. When they type on their keyboard the input goes into the notes window. By using the wikilinks syntax the user can create multiple notes and link them together For example, a note titled “Plato” would link to a note titled Aristotle if the text file reads “Plato was a student of

[Aristotle]“ and a note titled “Aristotle” exists. Sitting on the desk is a visual representation of the notes in the notes window in the form of a knowledge graph.

Each node in the knowledge graph will represent a text file, and each line connecting nodes will represent a reference within the text file to another note text as specified by the use of brackets. As the user edits notes in the notes window the knowledge graph will update to represent these changes. This interface is designed as a research tool so that as the user conducts research they can build up the knowledge graph to contain more and more information. The 3D knowledge graph representation will become more useful gradually over time as the network visualisation of information in notes expands and the visualisation becomes more complicated.

The user can manipulate the 3D knowledge graph visualisation with hand tracking and gesture visualisation. The user can pinch corners of the graph visualisation with two hands and drag outwards to expand the visualisation or grab with one hand to move and rotate the visualisation. The force of gravity applies to this visualisation so when it is dropped it will land on the table. If it is dropped off the table the visualisation will respawn on the table conveniently in front of the user.

The application also has a browser component which allows the user to do the research that they will be taking down in the notes application. Using a small text bar floating just above the user's keyboard they can search the internet for information. When the user hits the “enter” key in this text box, the application will spawn a browser window with the search text of the query that the user gave. These windows can each be positioned and resized as many times as the user wants.

Implementation of Redesigned Note Application With Volumetric Graph Interface.

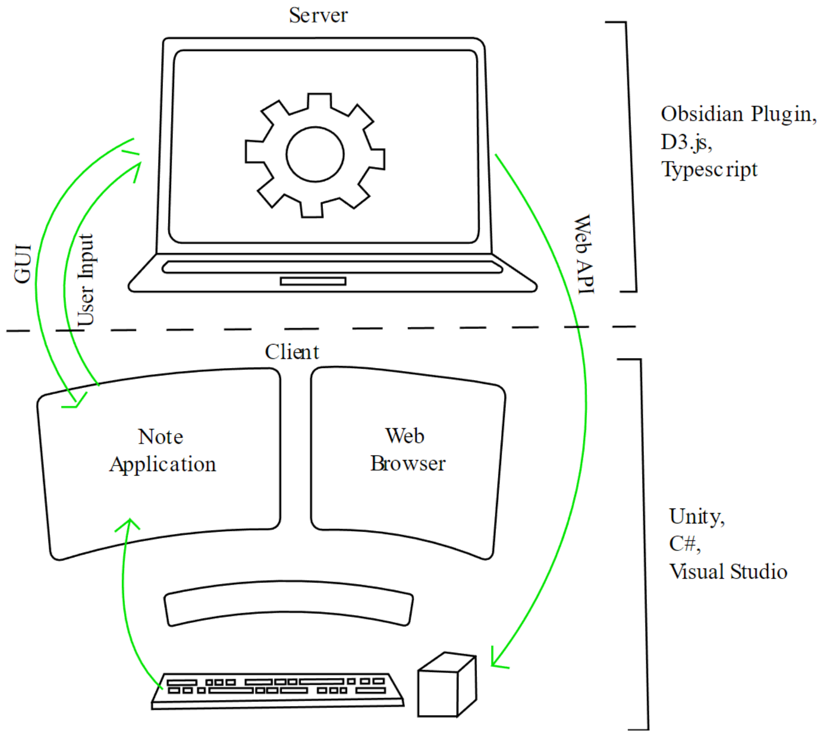

Figure 28. A diagram of the system's client server configuration.

System Architecture Overview: The application has two distinct parts to it: the client side which runs on the headset and the server side which runs on a remote computer. The client side is built with unity and runs in Visual Studio 2019. The server side runs obsidian with a special modified obsidian plugin which serves graph data over the internet to the client side via restful API. The client side is written in c# code while the server side is typescript code. Both the client and server side use a wide range of packages and tools to make their function possible. The client side i.e. Unity is primarily responsible for managing user input and interactions with content it generates, while the server side takes care of back end logic for parsing through note text and uncovering semantic relationships based on the content of the notes.

Client Server Model: This client server setup is useful for two reasons. First and foremost, it allows for two different languages to be used as there is a lack of 3D graph layout tools available for C# projects. Secondly, the two part setup of the application allows for much of the computationally intensive tasks of parsing text into a graph data structure and then running a force graph simulation off to a more powerful server computer so that the headset can run the server side application quickly.

At a fundamental level the server side runs the note taking application and custom backend logic that lays out the graph and the client side accesses the GUI via remote desktop and the graph data Via remote API. The client side simply displays the GUI, generates the graph as specified by the web api, and manages interactions in the virtual environment.

Description of GUI Components

Instructions Panel: Instructions appear in front of the user when they first open the application. This is a basic translucent panel which can be closed by hovering over it and pressing alt + w.

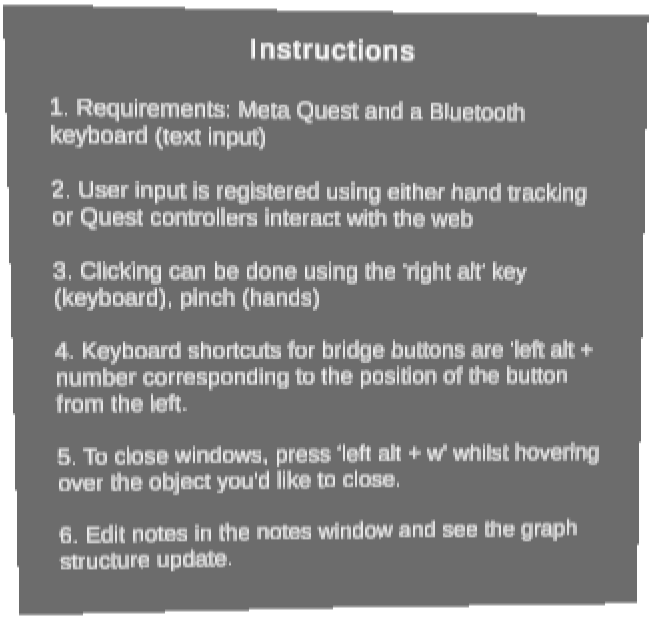

Figure 29. Screenshot of the instructions panel

Windows: Windows were implemented using the viewplex webview plugin for Android. This package implements a basic version of a web browser that can run in a Unity 3D scene. Further functionality was built on top of the base Vuplex Web Browser such as favoriting, refreshing, positioning, and resizing windows. The notes window exists within the personal proximic zone near the designed focal distance of the headset where the image is sharpest, which is important for reading text. The web browser components are set further back in the social space proximic zone signifying the content is from the public domain. Despite being pushed back, the headsets focus at this range is still strong and the user can scale the interface using the window tools or zoom in the web browser using ctrl + / - keys as they would normally on desktop to make the content clearer. The window is locked along the Z-axis depth, However it is free to be resized and repositioned along the x and y axis.

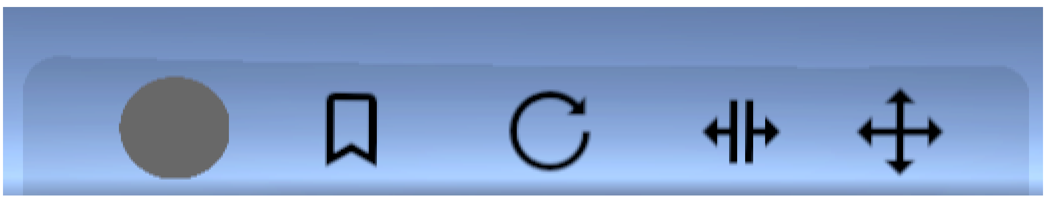

Figure 30. A screenshot of the window controls. Going from left to right the Window has an icon and 4 buttons. 1. Bookmark, 2. Refresh, 3 Resize and 4. Reposition.

Bridge: The “bridge” component is a small tab which hovers at the end of the users desk. The bridge has a text input field and buttons that allows the user to toggle various settings such as the pointer, raycasting and the debug window. There is also a button for resetting the simulation, which is useful for restarting hand tracking or regenerating the graph. The bridge’s text input field enables users to conveniently search the web and spawn browser windows into the virtual environment when the enter key is hit. The bridge also has a bookmarks bar which can hold the icons of tabs that have been bookmarked. A tab can be bookmarked by selecting the bookmark icon at the top of the browser tab. That tabs icon will be added to the bookmark bar and should the user wish to open this tab in the future they simply have to select the icon in the bookmark bar. The orientation of the bridge is locked to the user's head position so the user will always be able to clearly see this element.

Figure.31 Screenshot of the bridge component

The Status Indicators Tab: If the user looks directly up they can see the status indicators tab. Here they can see information about systems that are outside of their explicit control such as the battery percentage of the headset, the state of the keyboard tracking software and date and time.

Tracked Keyboard: The Meta tracked keyboard SDK integrates a physical keyboard into virtual environments using the cameras on the front of the meta quest headset to continuously track a physical keyboard in real time enabling users to see and interact with their keyboard within the virtual space. This results in an enhanced sense of presence and immersion. The movements of users' real-world keyboard and hands are translated into the virtual environment, allowing them to visually perceive their hands and the keyboard while immersed in a virtual environment. This integration creates a tactile typing experience, bridging the gap between the physical and virtual worlds. In this case the keyboard being used was an Apple magic keyboard. The Meta Quest headset handles keyboard input and passes it to the Unity application. From there custom C# logic had to be implemented to ensure that the keyboard input was being passed to the correct component in the application.

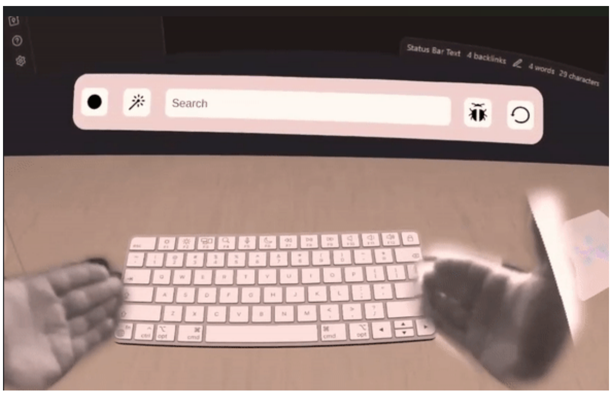

Figure 32. Screenshot of a tracked keyboard and hands passthrough.

Passthrough: The user can toggle the degree to which they can see their outside environment. Physical hands can be seen via passthrough when they are placed over the keyboard or the entire keyboard can be seen via passthrough with key label overlay for easy identification of keys. If the user toggles the "Toggle keyboard presentation" poke button then the keyboard presentation will switch from being a 3D model of a keyboard, to showing the keyboard through the cameras on the front of the device.

Custom Keys and Controls: The alt buttons on the keyboard have been customised to provide the keyboard with additional functionalities. The alt key on the keyboard functions the same way as the pinch gesture for selecting interface elements. When the user selects the alt key and a number key that corresponds to a button that button will be toggled. This was done in an attempt to allow the user more input modes and not make them take their hands off the keyboard and change the input method to hand tracking

- Alt: Select With Hand Pointer.
- Alt w: Close window
- Alt 1: Toggle Pointer
- Alt 2: Toggle Raycast
- Alt 3: Open Debug Window
- Alt 4: Reset Simulation

Poke Buttons: Poke buttons are buttons that the user toggles not by using the hand pointer and pinch visual but by pushing the button down. Poke buttons provide a more natural and intuitive interface for making selections. There are poke buttons for debug and reset as well as toggle keyboard layout from the 3D model presentation to the passthrough presentation. When pressed with hand tracking the poke button will provide visual feedback in the form of a colour change and audio feedback in the form of a "clack" sound.

Figure 33. Four poke buttons

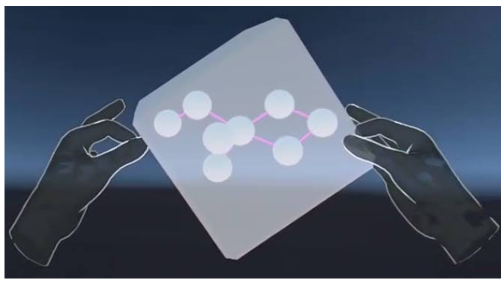

Figure 34. The graph interactable

Graph Object: The user can interact with a volumetric interface graph visualisation by means of hand tracking and gesture recognition. This interface is generated based on the relationship between notes in the notes application. Each graph is composed of nodes with wraparound text showing the name of the node & clearly visible purple edges. The node text tracks the user's head position so that the node name which wraps around the node sphere is always visible. Purple lines signify edges or relationships between notes which are represented by white spheres. Note names are only visible when the user is close enough to the nodes to be able to read the name text. The graphs function can be broken down into a few components.

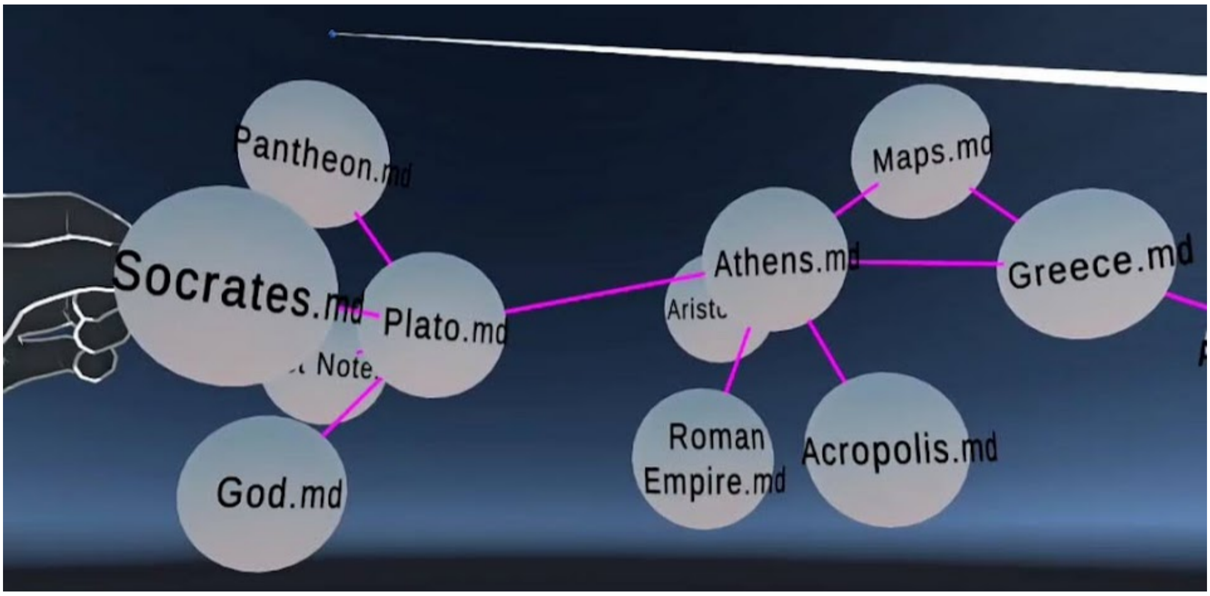

Figure 35. A close up of the graph structure showing node names

1) The Physics Rigidbody: This component controls the physics and interactions between the hands and graph objects. It can be thought of as the translucent cube encasing the nodes and edges of the graph. When dropped off the table the graph will respawn on the desk again so that the user can once again reach the graph object and manipulate it. Graphs can be pinched and held with one hand or scaled with two hands. The minimum size of the node is set to 1 unit and the maximum size of the graph is 15 units so that the graph is always clearly visible but never gets large enough to be difficult to manage. Physics such as gravity and collisions apply to the graph meaning when raised and dropped the graph will fall onto the desk.

2) The API Manager: The API manager Script is responsible for continuously interrogating the Server Web API feeding the application with node coordinates. The API Manager reads in these coordinates and if different to the current state of the graph object it will delete the current graph object and regenerate it according to the current scale of the graph as determined by the user's interactions with the Physics RigidBody Component. In order to do this the graph has to first be generated at world coordinates and then be transformed to the centre point of the physics rigidbody (so that it aligns with the Rigidbody) then scaled down by a factor of how many times the physics rigidbody fits within the bounds of the generated nodes and edges.

Process Json.cs

```csharp
void processJson(string _url)
{
    //Make Graph
    Graph newGraph = JsonUtility.FromJson<Graph>(_url);
    if (newGraph.Equals(graph) == false)
    {
        Destroy(graphGameObj);
        graph = newGraph;

        //Calculate graph size and cube size
        float graphWidth = Math.Abs(graph.boundingbox.x[1] - graph.boundingbox.x[0]);
        float graphHeight = Math.Abs(graph.boundingbox.y[1] - graph.boundingbox.y[0]);
        float graphDepth = Math.Abs(graph.boundingbox.z[1] - graph.boundingbox.z[0]);

        Vector3 graphSize = new Vector3(graphWidth, graphHeight, graphDepth);
        Vector3 cubeSize = bevelCube.bounds.size;

        //Make graphGameObj Cube
        graphGameObj = GameObject.CreatePrimitive(PrimitiveType.Cube);
        graphGameObj.transform.localPosition = new Vector3(0, 0, 0);
        float widestEdge = Mathf.Max(graphSize.x, graphSize.y, graphSize.z);
        graphGameObj.transform.localScale = new Vector3(widestEdge, widestEdge, widestEdge);
        graphGameObj.GetComponent<BoxCollider>().enabled = false;


        //Draw Nodes
        foreach (nodesList node in graph.nodes)
        {
            Debug.Log("Creating node in loop");
            createNodePrefab(node, nodeScaler);
        }

        foreach (linksList link in graph.links) { createLink(link); }

        //Downscale
        graphGameObj.transform.SetParent(bevelCube.transform);
        graphGameObj.transform.localScale = (bevelCube.bounds.size / 100) * 80;
        graphGameObj.transform.position = bevelCube.bounds.center;
        graphGameObj.GetComponent<MeshRenderer>().enabled = false;
    }

}
```

Snipped

Figure 36. *Json Manager Code*

The Notes Application: The notes application is being served to the client via remote desktop through a browser window. The browser window registers where the raycaster coming from the user's hand is hitting the virtual display and whether or not a pinch has been registered and sends this as input to the server running the application. In this way the user can use a browser window to manipulate an application running on a more powerful machine than the Quest headset. The note application that is running on the headset is called Obsidian. As this is an application that is not open sourced it was not possible to break the original 2D interface out into its distinct components such as search bar and toolbar and create a sense of informational hierarchy as previously discussed.

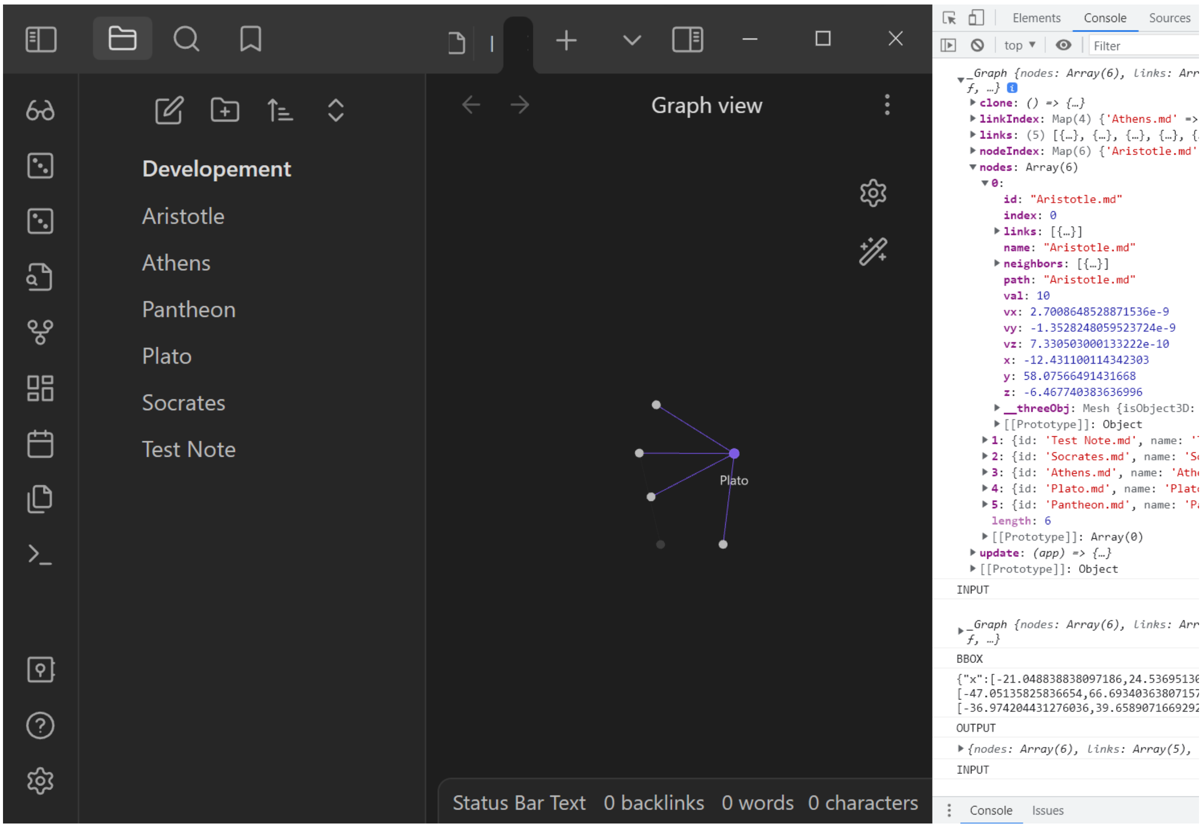

Figure 37. *Obsidian.md Open in Developer Mode*

Obsidian Plugin: The server side computer running the note taking application has been modified via a plugin. Plugins are ways for developers to create new features for the obsidian application by using the obsidian API to retrieve and modify note data. An obsidian Plugin was developed to generate the layout of a 3D Graph for the purposes of visualisation in Unity. The layout of the graph was generated by using the obsidian API to retrieve a graph data structure that represented the relationships between markdown files in the application. Then this data structure was visualised using d3.js' force directed graph algorithm. D3.js runs a simulation that applies opposing forces such as centre, repel and link forces to a network graph to get it to spread out across three dimensions. This is done iteratively with each iteration of the simulation called an "engine tick".

ForceGraph.ts

```
this.instance = ForceGraph3D()(this.rootHtmlElement)
            .graphData(this.getGraphData())
            .nodeLabel(
                (node: Node) => `<div class="node-label">${node.name}</div>`
            )
            .nodeRelSize(this.plugin.getSettings().display.nodeSize)
            .backgroundColor(rgba(0, 0, 0, 0.0))
            .width(width)
            .height(height)
            .onEngineTick(() => {
                iteration++;
                console.log("iteration " + iteration +": ");
                console.log(this.graph);
            });
```

Snipped

Figure 38. *Screenshot of D3.js code*

The Web API: was set up as part of the obsidian plugin so that the client unity application could access the graph layout data. This Web API constantly serves a graph data structure which updates on every engine tick. This way, the client can decide how often to update the graph gameobject as its data is constantly available over an http fetch or get request. The application is set to check if the graph data structure requires updating every 140 frames which is roughly equal to 2 seconds as the unity application typically runs at about 70 frames per second.

ForceGraph.ts

```
        const server = http.createServer((req: IncomingMessage, res: ServerResponse) => {
            res.writeHead(200, { "Content-Type": "application/json" });

            const simplifiedData = simplifyData(this.graph, this.instance.getGraphBbox());
            res.end(JSON.stringify(simplifiedData));
        });

        server.listen(3000, () => {
            console.log('Server listening on port 3000');
        });
```

Snipped

Figure 39. *Screenshot of Web API code*

Figure 40. Sample JSON Graph Layout Output (read left then right image)

Input Modalities: Although exploring the capabilities would have been an exciting challenge, the Meta Quest 2 headset which was used for this implementation does not offer this capability. Therefore the primary input method was hand tracking. The system also works with controller input. Hand tracking is an intuitive and implicit input method that compliments the use of a keyboard input as it keeps the hands free and visible. Controller input on the other hand can be more useful for precision tasks such as manipulating interfaces that are small or far away using pointers and raycasters.

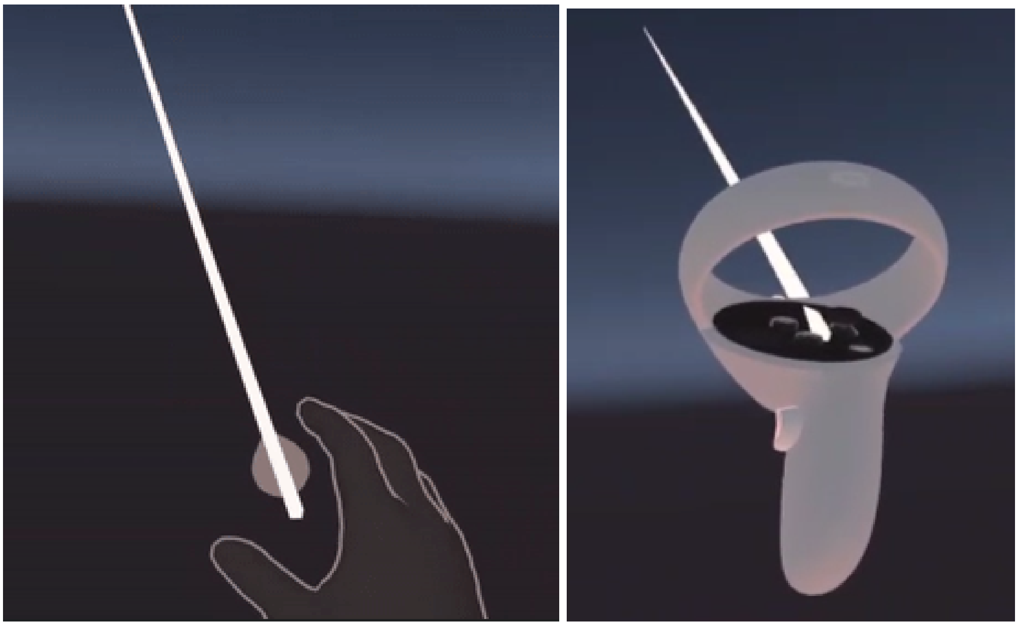

Figure. 41 Hand and controller pointer

Discussion

Design Consideration: The Graph

Initially two primary approaches seemed promising for the application:

- Geometric Transformations: This method begins with a 2D layout, transforming it into 3D through techniques like rotation, inflation, or projection.
- Force-directed Algorithms: These are algorithms that employ forces (attraction and repulsion) to achieve optimal graph layouts. Examples include:
    - Fruchterman-Reingold: uses repulsive forces between nodes and spring-like attractive forces between edges, ensuring a balanced layout with minimal overlaps.
    - Kamada-Kawai: Focuses on maintaining consistent edge lengths, producing a layout that mirrors actual graph distances.
    - Barnes-Hut: Known for its efficiency, this algorithm approximates forces between distant nodes, rendering it suitable for larger graph layouts.

While Unity is a powerful game development platform, hurdles were encountered when attempting to lay out a 3D graph within it. The primary challenge arose from the scarcity of tools optimised for real-time 3D graph layout and visualisation within Unity. Although 2D graph generators such as QuickGraph and MSAGL were available in C#, their capabilities did not extend to 3D graph layouts. Potential solutions included:

- **JavaScript Frameworks:** Several JavaScript libraries, notably A-Frame, d3.js, and NGraph, could generate 3D network graphs. However, the concern was about the integration of these frameworks with Unity.
- **Standalone Applications:** Gephi and Cytoscape are powerful tools, suitable for deep graph analysis and visualisation. The downside, however, is that they lack programmatic interfaces, making data import/export cumbersome and offering no straightforward path for Unity integration.
- **Unity's Physics Engine:** An initial idea was to employ Unity's physics capabilities to create a force-directed graph from scratch. Though promising at first, it proved to be a complex endeavour—a sentiment shared by other developers who documented their struggles online.

I ultimately selected the force-directed graph approach using custom force weights. This decision was influenced by the technical limitations that were encountered and the fact that the 2D application that was being redesigned also used a force directed graph layout, albeit a 2D version. Additionally, the dynamic nature of force directed graphs and their adaptability to real-time modifications, made them an apt choice. It was decided that D3.js was to be used, to simulate the three-dimensional node expansion and contraction. D3,js was used as it could be integrated into the obsidian plugin and was extensively documented, although an A frame implementation fits this criteria also.

Design Consideration - Platform:

Native:

- Pros: The application runs directly on the VR headset. This eliminates the need for any external wires, offering a more immersive experience without restrictions. Another advantage is the ability to host the application on various app stores, facilitating wider distribution and accessibility.
- Cons: Native applications require dedicated development and might necessitate periodic updates to stay compatible with newer hardware versions or operating system changes.

Desktop:

- Pros: Desktop applications generally have the benefit of tapping into the computational power of a dedicated PC, which could enable more sophisticated functionalities or visuals.
- Cons: They require a direct connection to a PC, which can impede the user experience. Wires can easily become tangled, pose a tripping hazard, or even obscure access to essential peripherals like the keyboard.

PWA (Progressive Web App):

- Pros: PWAs are known for their versatility, being accessible on a wide variety of devices.
- Cons: The primary limitation of PWAs in the VR context is the lack of control over the environment and, more specifically, the inability to manage a volumetric interface effectively.

WebXR:

- Pros: WebXR is a promising technology for developing virtual and augmented reality experiences on the web.
- Cons: Its current iteration doesn't support hand tracking, making interaction in a VR environment less intuitive. Moreover, the lack of tracked keyboard integration and passthrough functionality made it a poor choice for this project.

The final choice was to implement a native application as it allowed me to take advantage of tracking functionalities enabled by running the software directly on the headset, which would not have been the case otherwise.

Design Consideration - Game Engine:

When comparing the Unity and Unreal game engines, Unity emerged as the more fitting choice for the redesign. One of the distinguishing advantages of Unity is its support for scripting with C#. While Unreal offers visual scripting via Blueprints, they often don't provide the same depth and flexibility that C# scripting in Unity does, especially when developing intricate VR interactions like those required for this notes application.

6. Evaluation

Performance

The performance of the application is mostly constrained by the limited computing power available to the client-side unity application running on the meta quest headset. The Server side of the application does not cause performance issues as the computational power of a server can be scaled as required and is generally more than sufficient to run a virtual desktop and a notes application at the same time. The computational power of the meta quest headset on the other hand is significantly constrained by its hardware.

Frame Rate: The performance of the VR headset is crucial as it directly affects the smoothness of the virtual reality experience. If the processing power of the headset cannot keep up with the demands of the application, such as maintaining a frame rate of at least 75fps, it can lead to a decrease in performance. Most of the features of the application consume a fixed amount of computational resources regardless of how the user interacts with them (hand tracking, interface interactions, keyboard tracking etc). There are however two elements that can have a large variable impact on performance. These are 1) the number of open windows within the application and 2) generating graph objects at regular intervals.

The **number of open windows** within the application places computational strain on the system as WebView components work by simulating a browser capturing image textures from them and streaming them onto planes in 3D space, which is computationally expensive. This results in a steady decrease in the frame rate of the application as the number of open windows increases, potentially leading to a laggy and unresponsive user experience.

The limitations of the system were analysed by recording frame rates that were observed when different numbers of windows were spawned each with a set resolution of 720 x 480. This analysis shows that despite the potential for panoramic virtual displays, technical limitations still reduce the amount of displays that a user can render effectively in virtual space. Once 10 different browser tabs were opened the frame rate of the system fell consistently below 75fps, the minimum recommended frame rate for virtual displays, as examined earlier in the report. This is likely more tabs than any user would want to have open as they would struggle to arrange this many browser windows effectively.

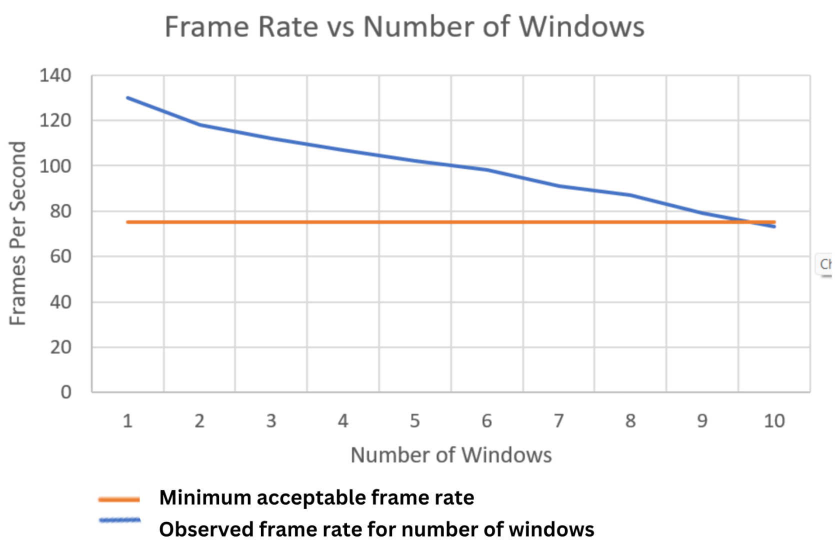

Figure 42. A plot showing decreasing frame rate as more windows are added

In future implementations, it would be beneficial for there to be the option to hide multiple browsers in one window using a tab system, as is common with 2D web browsers. Alternatively approaches for providing additional resources to displays in the user's direct line of sight may hold potential for improving perceived performance. These features fell outside the scope of this project

Generating graph objects is another computationally intensive task which requires an unusually high demand to be placed on the system at regular intervals as the graph updates itself every 140 frames. If the graph is too complex the next frame in the sequence can be delayed while the system generates and positions the required geometry. This leads to the image presented on the display stuttering which can cause nausea.

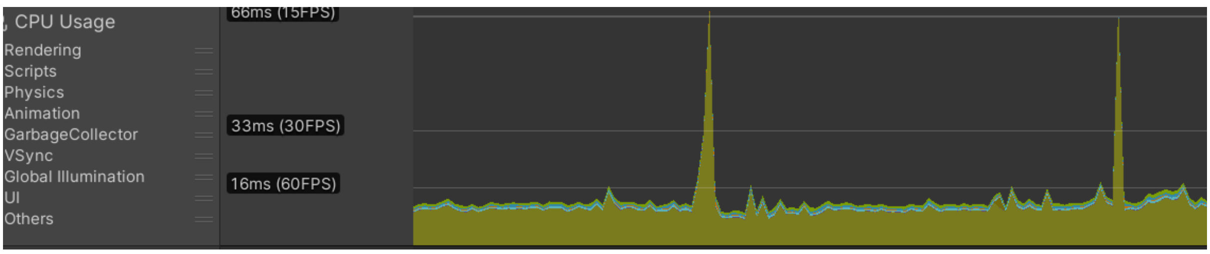

Figure 43. Unity profiler displaying regular drops in the frame rate

To measure and visualise this effect the unity profiler was used to analyse the performance of the application and provide detailed insights into aspects like CPU usage, memory allocation, and rendering statistics at the time of the jump between frames. This confirmed suspicions that the script responsible for interrogating the server API and generating the required geometry was halting the application momentarily while it ran. Once generated, transformations on the graph object were processed smoothly.

In future implementations, it would be wise to identify performance bottlenecks such as this that might occur, particularly concerning visualising data, which can be computationally expensive. In scenarios where a bottleneck is hit, asynchronous code execution techniques might provide a solution to the jarring effect of having a slow refresh rate.

Hand tracking and gesture recognition were found to be temperamental at times when parts of the hand were occluded. For instance, issues with hand tracking were encountered when the thumb was occluded from the cameras on the front of the headset by a wire or another part of the hand. This was a common occurrence which caused a nauseating mismatch between the presented positioning of the hand in VR and the user's proprioception. Likewise, keyboard tracking worked well when the keyboard was in sight of the cameras however when it was partially occluded, it was prone to jumping around in the environment as the headset mistakenly placed it at the wrong position in space.

Pointers worked for interacting with interface elements that could not be reached. The controllers provided greater accuracy, which was probably due to the motion sensors in the controller providing more accurate feedback to the headset than the computer vision algorithms for hand tracking implemented by Meta.

The graph application worked as expected. There was a slight but noticeable delay when an element was selected and when the interface updated to reflect the changes in the application state, although this was in the order of fractions of a second. Overall, the experience of interacting with an application that was originally designed for 2D displays was responsive and usable.

Utility - Revisiting Hierarchical Task Analysis

The revisited task analysis provides an evaluative framework to critically assess the completion and integration of specified functionalities within the VR Notes Application. Each task is marked with either a tick (✅) or an X (❌), offering a clear visual representation of successfully implemented features versus those yet to be realised or integrated (see appendix). This methodological approach not only offers a granular understanding of the application's current capabilities but also facilitates the identification of areas of improvement, potential bottlenecks, and user experience gaps.

Out of the 8 primary tasks, 6 have been fully realised, demonstrating a strong alignment with the project's core goals. However, areas of improvement emerge in the realms of note organisation and collaboration. The inability to assign unique identifiers to categories may impede the user's visual recognition, and the absence of note-moving and searching features may slow down the user's workflow. The complete omission of collaborative features further restricts the application's use cases. Future iterations of the application would benefit immensely from addressing these areas. The foundational aspects, like note creation, access, editing, and visualisation are facilitated.

Usability - Revisited Design Guidelines

The application did not undergo any formal user testing for usability. It was however designed with a comprehensive set of 20 design guidelines established to inform the redesign and development of computer work applications for HMDs. These guidelines were carefully crafted and outlined previously based on both theoretical foundations and practical considerations. These guidelines were continuously referenced and implemented in the actual development process. As a result, each guideline is reflected in one or more features within the application. From considering ergonomic content placement and utilising real-world physics to incorporating hand-tracking and gesture recognition capabilities, nearly every aspect of the developed VR environment demonstrates adherence to these guidelines. Please refer to the appendix for further details regarding all included features.

The purpose behind creating and implementing these guidelines was dual-faceted. Firstly, it served to standardise and streamline the development process, ensuring that the user experience remained at the forefront. Secondly, and perhaps more critically, the development of the VR notes application served as a tangible testing ground for these guidelines. By incorporating these design elements, the project aimed to enhance user experience while also focusing on validation and improvement. The application served as a testing ground to assess the practicality and efficacy of the guidelines, guaranteeing that they not only appear promising in theory but truly enhance an intuitive, immersive, and comfortable VR experience.

7. CONCLUSION

The growth in HMD adoption hints at a paradigm shift in HCI, spotlighting spatial computing's potential. This paper delineates the nuances of adapting traditional 2D computer interfaces for HMDs, providing evidence-based guidelines that cater to the unique capabilities of MR. Through a robust exploration of ergonomic considerations, the development of a three-component design approach, and the successful implementation of a demo application, this work serves as a foundational blueprint for those venturing into the design of MR applications. It underscores the significance of leveraging pre-existing 2D mental models and frameworks, offering a seamless transition into the realm of MR, with emphasis on user comfort and practicality. Importantly, it recognizes and addresses challenges, from ergonomic considerations to the intricate design demands of HMDs. As HMD technology continues to garner attention and investment, the methodologies and insights presented here pave the way for a future where MR is not just a supplementary tool but a transformative force in productivity and computer work. This research not only serves as a compass for designers and developers in the MR sphere but also as a call for more rigorous, application-driven studies to bridge any existing gaps.

This dissertation comprises two primary segments, with the initial phase centred on extensive research. This part begins by clarifying VR terminology to dispel common misconceptions and progresses to provide a technical overview of current VR headsets, laying a foundation for subsequent design considerations. The research narrows its focus on the application of VR for computer-related tasks, analysing existing studies and emphasising design suggestions that limit content placement areas. Recognizing that limitless spatial content can hinder user-friendly design, especially for extended use, the study emphasises the need for specific content zones, driven by technical and ergonomic considerations. Within these defined zones, the research explored content types best suited for adaptation from 2D settings and identified new components optimised for the spatial features of HMDs. A tri-component framework was introduced for the redesign process, encompassing 1) windows to incorporate and adapt 2D content, benefitting from depth based information heirarchy, 2) volumetric visualisations or 3D objects to augment the user experience, and 3) meticulously crafted virtual environments that guide user interactions in line with the designer's intent. The research phase culminates with 20 pivotal guidelines for transitioning computer work to HMDs, aiding designers in proficiently utilising the three components to reimagine their work applications for an immersive environment.

The second part of the report focuses on the implementation of a redesigned computer work application based on the conducted research. The aim was to implement and test the concepts discovered during the research phase such as the content zones, the suggested 3 part component framework and the design guidelines for how to approach the redesign of these components. In doing so foundational knowledge about the process of redesigning a 2D app for use in HMDs was contributed. The application focussed on the task of conducting research and taking notes and approached the problem from a novel visualisation standpoint.

The application provides a dynamic and user-focused interface for consolidating and visualising research. By integrating text input with an expandable 3D knowledge graph, the application allows for the creation, linking, and visualisation of notes in a comprehensible and engaging format. Through features like wiki link syntax and real-time editing, research becomes more efficient and interconnected. The use of hand tracking and gesture visualisation technology adds an extra layer of interactivity, permitting users to manipulate their visualisations directly, creating a more immersive experience. The inclusion of natural, intuitive actions like expansion, rotation, and relocation of the visualisation enhances user experience and makes information engagement more intuitive. The redesign was measured by its performance but also by how many of the predefined user tasks it facilitated. The design guidelines were revisited to ensure that the application was successful in implementing all of the guidelines for the redesign of a computer work application for a HMD.

Future work should attempt to investigate various input modalities for these systems such as gesture recognition, eye tracking, voice-activated controls or brainwave sensors. Any future work aiming to redesign a 2D computer work application for use in a HMD could follow the content zones, three-part component framework, discovered design guidelines and methodology described in this report and have a strong foundation for the redesign of their application.

8. References

Adam, J.A. (1993) ‘Virtual reality is for real’, IEEE Spectrum, 30(10), pp. 22–29. doi:10.1109/6.237580.

Alexandrovsky, D. et al. (2020) ‘Examining design choices of questionnaires in VR user studies’, Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems [Preprint]. doi:10.1145/3313831.3376260.

Alexandrovsky, D., Putze, S. and Schülke, A. (2021) ‘Towards Low-burden Responses to Open Questions in VR’,. (2021). Towards Low-burden Responses to Open Questions in VR. doi:10.48550/arxiv.2104.12020.

Alger, M. (2016) ‘HMD resolution and maximum depth perception.’ Available at: http://mikealgermovingimage.tumblr.com/post/127113260256/hmd-resolution-an d-maximum-depth-perception.

Ammann-Reiffer, C., Kläy, A. and Keller, U. (2022) ‘Virtual reality as a therapy tool for walking activities in pediatric neurorehabilitation: Usability and User Experience Evaluation’, 10(3). doi:10.2196/38509.

Anselin, L. and Rey, S.J. (2012) ‘Spatial econometrics in an age of cybergiscience’, International Journal of Geographical Information Science, 26(12), pp. 2211–2226. doi:10.1080/13658816.2012.664276.

Arbeláez-Estrada, J.C. and Osorio-Gómez, G. (2013) ‘Augmented reality application for Product Concepts Evaluation’, Procedia Computer Science, 25, pp. 389–398. doi:10.1016/j.procs.2013.11.048.

Billinghurst, M. and Kato, H. (2002) ‘Collaborative augmented reality’, Communications of the ACM, 45(7), pp. 64–70. doi:10.1145/514236.514265.

Billinghurst, M., Clark, A. and Lee, G. (2015) ‘A survey of Augmented Reality’, A Survey of Augmented Reality [Preprint]. doi:10.1561/9781601989215.

Blissing, B., Bruzelius, F. and Eriksson, O. (2022) ‘The effects on driving behavior when using a head-mounted display in a dynamic driving simulator’, ACM Transactions on Applied Perception, 19(1), pp. 1–18. doi:10.1145/3483793.

Boletsis, C. and Kongsvik, S. (2019) ‘Text input in virtual reality: A preliminary evaluation of the drum-like VR keyboard’, Technologies, 7(2), p. 31. doi:10.3390/technologies7020031.

Bowman, D.A. (2005) ‘Workshops’, IEEE Proceedings. VR 2005. Virtual Reality, 2005. [Preprint]. doi:10.1109/vr.2005.1492817.

Bowman, D.A. (2020) ‘Embracing physical keyboards for virtual reality’, Computer, 53(9), pp. 9–10. doi:10.1109/mc.2020.3004605.

Bowman, D.A. et al. (2008) ‘3D user interfaces: New directions and Perspectives’, IEEE Computer Graphics and Applications, 28(6), pp. 20–36. doi:10.1109/mcg.2008.109.

Budziszewski, P. et al. (2016) ‘Workstations for people with disabilities: An example of a virtual reality approach’, Journal of Safety and Ergonomics, 22(3), pp. 367–373. doi:10.1080/10803548.2015.1131069.

Butow, E. (2007) User interface design for mere mortals: A hands-on guide to user interface design ; software-independent approach. Boston: Addison-Wesley.

Carter, M. and Egliston, B. (2021) ‘A critical future of virtual reality: All work and no play’, AoIR Selected Papers of Internet Research [Preprint]. doi:10.5210/spir.v2021i0.12152.

Chen, J.Y. and Thropp, J.E. (2007) ‘Review of low frame rate effects on human performance’, IEEE Transactions on Systems, Man, and Cybernetics, 37(6), pp. 1063–1076. doi:10.1109/tsmca.2007.904779.

De Boeck, J., Raymaekers, C. and Coninx, K. (2008) ‘A tool supporting model based user interface design in 3D virtual environments’, Proceedings of the Third International Conference on Computer Graphics Theory and Applications, pp. 367–375. doi:10.5220/0001097703670375.

Earnshaw, R.A., Gigante, M.A. and Jones, H. (1993) Virtual reality systems. London, England: Academic Press.

Gong, L. et al. (2020) ‘Interaction design for multi-user Virtual Reality Systems: An automotive case study’, Procedia CIRP, 93, pp. 1259–1264. doi:10.1016/j.procir.2020.04.036.

Grout, C. et al. (2015) ‘Reading text in an immersive head-mounted display’, Proceedings of the 15th New Zealand Conference on Human-Computer Interaction [Preprint]. doi:10.1145/2808047.2808055.

Guo, J. et al. (2018) ‘Subjective and objective evaluation of visual fatigue caused by continuous and discontinuous use of hmds’, Society for Information Display, 27(2), pp. 108–119. doi:10.1002/jsid.750.

Human interface guidelines Apple (2023) *Apple Developer Documentation*. Available at: https://developer.apple.com/design/human-interface-guidelines (Accessed: 18 August 2023).

Hoffmann, H., Stefani, O. and Patel, H. (2006) ‘Extending the desktop workplace by a portable VR system’, International Journal of Human-Computer Studies, 64(3), pp. 170–181. doi:10.1016/j.ijhcs.2005.08.003.

John, M. et al. (1999) ‘Workspace control diagrams and head-mounted displays as alternatives to multiple monitors in information-rich environments’, Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 43(5), pp. 438–442. doi:10.1177/154193129904300510.

Kim, E. and Shin, G. (2021) ‘User discomfort while using a virtual reality headset as a personal viewing system for text-intensive office tasks’, Ergonomics, 64(7), pp. 891–899. doi:10.1080/00140139.2020.1869320.

Knierim, P. et al. (2018) ‘Physical keyboards in virtual reality’, Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems [Preprint]. doi:10.1145/3173574.3173919.

Lanier, J. and Biocca, F. (1992) ‘An insider’s view of the future of virtual reality’, Journal of Communication, 42(4), pp. 150–172. doi:10.1111/j.1460-2466.1992.tb00816.x.

LaViola, J.J., Bowman, D.A. and Kruijff, E. (no date) 3D user interfaces: Theory and practice. Addison-Wesley.

Lee, E.-S. and Shin, B.-S. (2021) ‘A flexible input mapping system for next-generation virtual reality controllers’, Electronics, 10(17), p. 2149. doi:10.3390/electronics10172149.

Liao, W., Hou, D. and Jiang, W. (2019) ‘An approach for a spatial data attribute similarity measure based on granular computing closeness’, Applied Sciences, 9(13), p. 2628. doi:10.3390/app9132628.

Lu, X. et al. (2020) ‘Exploration of hands-free text entry techniques for virtual reality’, 2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR) [Preprint]. doi:10.1109/ismar50242.2020.00061.

McMahan, R.P. et al. (2012) ‘Evaluating display fidelity and interaction fidelity in a virtual reality game’, IEEE Transactions on Visualization and Computer Graphics, 18(4), pp. 626–633. doi:10.1109/tvcg.2012.43.

Molina, J.P. et al. (2003) ‘Bridging the gap: Developing 2D and 3D user interfaces with the ideas methodology’, Interactive Systems. Design, Specification, and Verification, pp. 303–315. doi:10.1007/978-3-540-39929-2_21.

Moraes, A.N. et al. (2021) ‘The role of physiological responses in a VR-based sound localization task’, IEEE Access, 9, pp. 122082–122091. doi:10.1109/access.2021.3108446.

Nash, E.B. et al. (2000) ‘A review of presence and performance in Virtual Environments’, International Journal of Human-Computer Interaction, 12(1), pp. 1–41. doi:10.1207/s15327590ijhc1201_1.

Nichols, S. (1999) ‘Physical ergonomics of virtual environment use’, Applied Ergonomics, 30(1), pp. 79–90. doi:10.1016/s0003-6870(98)00045-3.

Nilsson, N.C. et al. (2018) ‘Natural walking in virtual reality’, Computers in Entertainment, 16(2), pp. 1–22. doi:10.1145/3180658.

Pausch, R., Proffitt, D. and Williams, G. (1997) ‘Quantifying immersion in virtual reality’, 24th annual conference on Computer graphics and interactive techniques; - SIGGRAPH ’97. doi:10.1145/258734.258744.

Pavanatto, L. et al. (2021) ‘Do we still need physical monitors? an evaluation of the usability of AR virtual monitors for productivity work’, IEEE VR and 3D User Interfaces. doi:10.1109/vr50410.2021.00103.

Peli, E. (1990) ‘Contrast in complex images’, Journal of the Optical Society of America A, 7(10), p. 2032. doi:10.1364/josaa.7.002032.

Pirker, J. et al. (2020) ‘Virtual reality in computer science education: A systematic review’, 26th ACM Symposium on Virtual Reality Software and Technology [Preprint]. doi:10.1145/3385956.3418947.

Polys, N.F., Kim, S. and Bowman, D.A. (2007) ‘Effects of information layout, screen size, and field of view on user performance in information-rich virtual environments’, 18(1), pp. 19–38. doi:10.1002/cav.159.

Pupillary distance (PD) (2020) *Pupillary distance - By qualified opticians | Glasses DirectTM*. Available at: https://www.glassesdirect.co.uk/eye-health (Accessed: 18 August 2023).

Royston, S. (2016) ‘A VR Environment for Interactive Network Vis’, *Scivis 2016* . doi:1604.08239.

Richa, M. et al. (2022) ‘High-level power estimation techniques in Embedded Systems Hardware: An overview’, The Journal of Supercomputing, 79(4), pp. 3771–3790. doi:10.1007/s11227-022-04798-5.

Sauer, Y. et al. (2022) ‘Assessment of consumer VR-headsets’ objective and subjective field of view (FOV) and its feasibility for visual field testing’, Virtual Reality, 26(3), pp. 1089–1101. doi:10.1007/s10055-021-00619-x.

Sherman , W.R. and Craig, A.B. (2018) ‘Understanding virtual reality’, The Morgan Kaufmann Series in Computer Graphics, pp. 4–10. doi:10.1016/c2013-0-18583-2.

Slater, M. and Steed, A. (2000) ‘A virtual presence counter’, Presence: Teleoperators and Virtual Environments, 9(5), pp. 413–434. doi:10.1162/105474600566925.

Spreij, L.A. et al. (2020) ‘Feasibility and user-experience of VR in neuropsychological assessment following stroke’, Neuropsychological Rehabilitation, 32(4), pp. 499–519. doi:10.1080/09602011.2020.1831935.

Studer, S. and Hofer, B. (2019) ‘Replication of the question-based spatial computing approach – experiences and suggestions for further developments’, GI_Forum, 1, pp. 40–53. doi:10.1553/giscience2019_01_s40.

Tanriverdi, V. and Jacob, R.J.K. (2001) ‘Vrid’, Proceedings of the ACM symposium on Virtual reality software and technology [Preprint]. doi:10.1145/505008.505042.

Thoondee, K.D. and Oikonomou, A. (2017) ‘Using virtual reality to reduce stress at work’, 2017 Computing Conference [Preprint]. doi:10.1109/sai.2017.8252142.

VR Interface Design Pre-Visualisation Methods (2015). YouTube. 4 October. Available at: https://www.youtube.com/watch?v=id86HeV-Vb8&t=355s (Accessed: 18 August 2023).

Walker, J. (2017) Improving everyday computing tasks with HMDs [Preprint]. doi:10.37099/mtu.dc.etdr/388.

Wang, J. et al. (2023) ‘Effect of frame rate on user experience, performance, and simulator sickness in virtual reality’, IEEE Transactions on Visualization and Computer Graphics, 29(5), pp. 2478–2488. doi:10.1109/tvcg.2023.3247057.

Wang, Yahui et al. (2021) ‘Investigating the performance of gesture-based input for mid-air text entry in a virtual environment’, Sensors, 21(5), p. 1582. doi:10.3390/s21051582.

Weiss, P.L. and Jessel, A.S. (1998) ‘Virtual reality applications to work’, Work, 11(3), pp. 277–293. doi:10.3233/wor-1998-11305.

Wexelblat, A. (1993) ‘Virtual reality: Applications and explorations’, Choice Reviews Online, 31(05). doi:10.5860/choice.31-2732.

Win, L.L. et al. (2022a) ‘Effectiveness on training method using VR and AR applications in Automobile Engine Assembly’, ASEAN Engineering Journal, 12(4), pp. 83–88. doi:10.11113/aej.v12.18009.

Witmer, B.G. and Singer, M.J. (1998) ‘Measuring presence in virtual environments: A presence questionnaire’, Presence: Teleoperators and Virtual Environments, 7(3), pp. 225–240. doi:10.1162/105474698565686.

XR Design Theory and Practice for Digital Eyewear (2020). YouTube. 7 February. Available at: https://www.youtube.com/watch?v=4o__z7aPlMw&t=12s (Accessed: 18 August 2023).

Ye, J. and Hua, K.A. (2013) ‘Exploiting depth camera for 3D spatial relationship interpretation’, Proceedings of the 4th ACM Multimedia Systems Conference [Preprint]. doi:10.1145/2483977.2483998.

Yildirim, C. and Osborne, E. (2020) ‘Text entry in virtual reality: A comparison of 2D and 3D keyboards’, HCI International 2020 Virtual and Augmented Reality, pp. 450–460. doi:10.1007/978-3-030-59990-4_33.

Yoshimura, A. and Borst, C.W. (2020) ‘Evaluation of headset-based viewing and desktop-based viewing of remote lectures in VR’, ACM Symposium on VR Software and Technology. doi:10.1145/3385956.3422124.

You, C.-W. et al. (2021) ‘The first workshop on multiple input modalities and sensations for VR/AR Interactions’, Proceedings of the 2021 ACM International Symposium on Wearable Computers. doi:10.1145/3460418.3479269.

‘Virtual reality application in engineering assembly education: A Review’ (2018) Journal of Engineering and Science Research, 2(6), pp. 17–23. doi:10.26666/rmp.jesr.2018.6.4.

9. Appendix

Common Computer Tasks

Presentations	Training	Location Finding	Calculation
Money Making	Word Processing	Finance	Email
Reference	Reading	File Management	Contacts
Video Creation	Instant Message	DataAnalysis/Vis	Shopping
Games	Schedule Planning	Watching Video	Development
Note Taking	Computer Management	Listening to Audio	Capturing Pictures/Video

Scenarios

Scenario 1

- Name: Sarah Mitchell
- Age: 32
- Occupation: Research Analyst at a Biotech firm
- Intentions: To effectively use the VR notes application for her extensive research work, which involves documenting multiple experiments, observations, and cross-referencing earlier notes.
- Desires:
    - A seamless transition from her traditional 2D note-taking application to the new VR platform.
    - The ability to visualise connections between different notes and experiments in an interactive 3D space.
    - Efficient tools for organisation and categorization, ensuring quick access to her extensive database of research.
    - A distraction-free environment, where she can deeply focus on her research notes.
- Challenges Encountered:
    - Initially overwhelmed by the vast spatial canvas of the VR platform.
    - Difficulty in navigating between nodes in the volumetric interface.
    - Occasional discomfort due to extended VR usage during long research sessions.
    - Adapting to the spatial representation of accumulated knowledge, compared to the traditional folder-based organisation she's accustomed to.

Scenario 2

- Name: Raj Patel
- Age: 27
- Occupation: Product Manager at a Tech Startup
- Intentions: To utilise the VR notes application as a collaborative tool for brainstorming and project planning with his remote team.
- Desires:
    - An intuitive space where he can visualise, edit, and annotate notes.
    - Features that support multimedia attachments, like images or videos, within the VR notes space.
    - A way to quickly switch between different projects or topics during a single VR session.
- Challenges Encountered:
    - Occasional lag in real-time, leading to disruptions during sessions.
    - Difficulties in establishing a common frame of reference between window and graph.
    - Adapting to the spatial representation of accumulated knowledge, compared to the traditional folder-based organisation she's accustomed to.

Task Analysis

1. Start VR Notes Application

    1.1. Power on VR headset.✅

    1.2. Adjust headset for comfort.✅

    1.3. Navigate to the application dashboard.✅

    1.4. Launch VR notes application. ✅

2. Accessing Existing Notes ✅

    2.1. Use hand gestures to navigate to the 'My Notes' section. ✅

    2.2. Scroll through a list of existing notes. ✅

        2.2.1. Use pinch gesture to zoom in on a specific note's details.✅

        2.2.2. Use swipe gesture to navigate between pages of notes.✅

    2.3. Select a note for a detailed view.✅

3. Creating a New Note✅

    3.1. Navigate to the 'Create' option.✅

    3.2. Initiate a new blank canvas.✅

    3.3. Input text using a tracked keyboard.✅

        3.3.1. Type out content.✅

        3.3.2. Use shortcut keys for text formatting (bold, italic, underline).✅

    3.4. Save the note.✅

        3.4.1. Choose a category or folder.✅

        3.4.2. Name the note for easy retrieval.✅

4. Visualising Note Connections ✅

    4.1. Navigate to the 'Visualisation' tab.✅

    4.2. Choose the '3D Knowledge Graph' option.✅

    4.3. View notes as nodes and connections as links.✅

- 4.4. Interact with the graph.✅
    - 4.4.1. Zoom in/out to view details.✅
    - 4.4.2. Rotate graph for different perspectives.✅
    - 4.4.3. Select specific nodes to view connected notes.✅

5. Editing a Note✅
    - 5.1. Open the desired note.✅
    - 5.2. Navigate to the 'Edit' mode.✅
    - 5.3. Make necessary changes.✅
    - 5.4. Save the edited note.✅

6. Organising Notes❌
    - 6.1. Create categories or folders.✅
        - 6.1.1. Name the category.✅
        - 6.1.2. Assign a colour or symbol for identification.❌
    - 6.2. Move notes to specific categories.❌
    - 6.3. Search for specific notes using keywords or dates.

7. Sharing and Collaborating❌
    - 7.1. Choose the 'Share' option on a specific note.❌
    - 7.2. Input recipient's VR username.❌
    - 7.3. Add comments or instructions for the recipient.❌
    - 7.4. Send the note.❌

8. Exiting the Application✅
    - 8.1. Navigate to the application's main menu.✅
    - 8.2. Choose 'Exit' or 'Log Out'.✅
    - 8.3. Remove VR headset.✅

Technical Feasibility Analysis for Redesigning a Notes Application for VR

Hardware Requirements

VR Headset:
- Considerations: High-resolution display for clear text, ergonomic design for prolonged use, and low latency for smooth interactions.
- Analysis: Most modern VR headsets meet these criteria. However, ensuring compatibility across various brands might be a challenge.

Input Mechanisms:
- Considerations: Need for a physical keyboard tracking system or a virtual keyboard interface.
- Analysis: While many VR setups allow for hand tracking, integrating real-world objects like keyboards seamlessly could pose a challenge.

Software Requirements

VR Platform:
- Considerations: Selecting a VR platform or engine to develop on, such as Unity or Unreal Engine.
- Analysis: Both major engines support VR development, but choosing one might depend on ease of implementation and flexibility for this specific application type.

Integration with Existing 2D Notes Apps:
- Considerations: Seamless transition and synchronisation between 2D and VR platforms.
- Analysis: Integration might require robust APIs and can become complex if multiple 2D note platforms are considered.

User Interface and Interaction

3D Graph Visualisation:
- Considerations: Designing a responsive, interactive 3D knowledge graph without overwhelming users.
- Analysis: Given the graphical capabilities of current VR platforms, this is achievable, but optimization will be key to prevent performance issues.

Text Display and Editing:
- Considerations: Legibility of text, ease of editing, and formatting in VR.
- Analysis: Text rendering in VR is challenging due to resolution constraints, but with careful design choices, it's feasible.

Features Implemented

- Guideline 1 - Notes window fixed at designed focus distance of headset (2m)
- Guideline 13, 16 - Proxemics Displayed through positioning of personal and personal information in respective bounds
- Guideline 20, 19 - Hand tracking and gesture recognition applied to object manipulation (pinch & drag)
- Guideline 16 - Content placed in ergonomic content zones (display & workbench zones)
- Guideline 8 - Customise Keyboard Input, Alt keys function as pinch gesture & alt + number keys function as buttons.
- Guideline 12 - Real world Physics (Gravity & Collisions)
- Guideline 19, 7 - Intractable Poke Buttons with feedback
- Guideline 15 - Passthrough Implemented over Keyboard and Hands
- Guideline 3, 13 - Use of depth to establish hierarchy (Tools closest then notes application then browser tabs)
- Guideline 2, 9, 4 - Windows Anchored to scene and curved to match the users Field of view.
- Guideline 9 **-** Windows orientation track users position
- Guideline 19, 17 - Dynamic lighting techniques to generate real time shadows
- Guideline 4 **-** No 3D text, Large text size for legibility
- Guideline 5 **-** Use of pointers to expand and reposition content
- Guideline 6 - Wide horizontal canvas for window placement.
- Guideline 7 Tactile Feedback gap addressed with Keyboard Feedback.
- Guideline 10, 7 - Visual cues and auditory signals used on poke button press
- Guideline 9 - Consider user fatigue by placing a keyboard and 3D object on a desk where hands can rest.
- Guideline 10 & 11 - Custom Gestures not used.
- Guideline 12 - 3D model integrated in interactive graph visualisation
- Guideline 13 - Scale and distance balanced in small close graph design to encourage closer inspection and manipulation.
- Guideline 14 - Environment kept smooth and simple with dark tones for comfort and simple office shapes.
- Guideline 15 - WIndow backdrop is made translucent to fit in with the environment.
- Guideline 18 - Keyboard interaction allows for left or right handed alt keys
- Guideline 19 Proxemics considered when placing graph interface at easily manipulable distance.
- Guideline 20 - Hand tracking visualisation helps to ground the user in the space

Examples of Industry Application of 3 Part Framework

Education

- Task: Lecture Preparation
- Occupation: Educator/Professor
- Primary Display: PowerPoint slides or course material
- 3D Model: A dynamic 3D model of the topic being covered, like the solar system
- Benefit: Allows the educator to manipulate and understand the topic in-depth, improving teaching quality.

Healthcare

- Task: Diagnosis Review
- Occupation: Radiologist
- Primary Display: Patient medical history and notes
- 3D Model: A 3D rendering of the patient's MRI or CT scan
- Benefit: Provides a more holistic view of the patient's condition, enhancing diagnostic accuracy.

Engineering

- Task: Component Design
- Occupation: Mechanical Engineer
- Primary Display: CAD software interface
- 3D Model: The component being designed in 3D, with stress points highlighted
- Benefit: Facilitates better understanding and refinement of the design, enhancing product reliability.

Architecture and Design

- Task: Building Layout Design
- Occupation: Architect
- Primary Display: Blueprint and design software
- 3D Model: A miniaturised 3D model of the building layout
- Benefit: Allows for a tangible, holistic view of the design, fostering better spatial planning.

Real Estate

- Task: Property Showcase
- Occupation: Real Estate Agent
- Primary Display: Property details, pricing, and documentation
- 3D Model: A walk-through model of the property
- Benefit: Enhances client presentation, aiding in a faster and more informed buying decision.

Retail and Marketing

- Task: Product Launch Strategy
- Occupation: Marketing Specialist
- Primary Display: Marketing campaign drafts and analytics
- 3D Model: The new product in 3D, with interactive hotspots for features
- Benefit: A more immersive understanding of the product, guiding more effective campaign strategies.

Construction

- Task: Infrastructure Review
- Occupation: Civil Engineer
- Primary Display: Infrastructure details and project management timelines
- 3D Model: A sectioned 3D model of the infrastructure, with layers like plumbing and electrical highlighted.
- Benefit: A more comprehensive review, ensuring a safer and more efficient construction process.

Tourism

- Task: Tour Package Design
- Occupation: Travel Agent
- Primary Display: Itinerary, pricing, and bookings
- 3D Model: Interactive 3D snippets of key attractions in the tour package
- Benefit: Enhances the package's appeal to clients, offering a "sneak peek" and driving sales.