Understanding Hand Interactions and Mid-Air Haptic Responses within Virtual Reality and Beyond

by

Rory Clark

Doctor of Engineering - Digital Media

Submitted at/Awarded by Bournemouth University

Centre for Digital Entertainment

September 2021

Copyright Statement

Acknowledgements

Throughout my studies, writing of this thesis, and placement, I have received help and guidance from numerous people. I would first like to thank my academic supervisor, Dr. Feng Tian, who has helped and assisted me across both my EngD journey and prior work. Thank you for pushing me to take this amazing opportunity and experience.

I would like to give considerable thanks to everyone at Ultraleap who I've both worked with and received support from over the past three years. Be it through study design, programming, or project management, your help has been greatly appreciated. To Adam Harwood and Pete Nancollis, thank you both for your guidance in helping to manage and review my work. Without the repeated invaluable help of David Beattie, the data analysis throughout this work would be nothing of the sort. Thanks to Pip Turner for his help throughout the open-sourcing of PlayRecorder, ensuring it was of high quality. I would also like to thank Naceur Abdouni for their repeated help and feedback over the years, especially throughout the first project. The first projects' user study would not have been achieved without the help of Dario Pittera, thank you once again.

The repeated help, support, and communication from Mike Board, Zoe Leonard, Sarah Parry, and Becca Knight at the Bournemouth and Bath Centre for Digital Entertainment (CDE) cannot be understated. Thank you all for the aid you have provided over these past four years. Thank you to the CDE for providing funding throughout my journey with the EPSRC grant EP/L016540/1.

Lastly, I would not have been able to complete this work without the support of my friends and family. My gratitude especially goes to Peach for being there throughout this journey, for your love and support along these past few crazy years.

September 2021

Rory Clark

Abstract

Hand tracking has long been seen as a futuristic interaction, firmly situated into the realms of sci-fi. Recent developments and technological advancements have brought that dream into reality, allowing for real-time interactions by naturally moving and positioning your hand. While these developments have enabled numerous research projects, it is only recently that businesses and devices are truly starting to implement and integrate the technology into their different sectors. Numerous devices are shifting towards a fully self-contained ecosystem, where the removal of controllers could significantly help in reducing barriers to entry. Prior studies have focused on the effects or possible areas for implementation of hand tracking, but rarely focus on the direct comparisons of technologies, nor do they attempt to reproduce lost capabilities.

With this prevailing background, the work presented in this thesis aims to understand the benefits and negatives of hand tracking when treated as the primary interaction method within virtual reality (VR) environments. Coupled with this, the implementation and usage of novel mid-air ultrasound-based haptics attempt to reintroduce feedback that would have been achieved through conventional controller interactions. Two unique user studies were undertaken, testing core underlying interactions within VR that represent common instances found throughout simulations. The first study focuses on the interactions presented within 3D VR user interfaces, with a core topic of buttons. While the second study directly compares input and haptic modalities within two different fine motor skill tasks. These studies are coupled with the development and implementation of a real-time user study recording toolkit, allowing for significantly heightened user analysis and visual evaluation of interactions. Results from these studies and developments make valuable contributions to the research and business knowledge of hand tracking interactions, as well as providing a uniquely valuable open-source toolkit for other researchers to use.

This thesis covers work undertaken at Ultraleap over varying projects between 2018 and 2021.

Contents

List of Figures

List of Tables

List of Publications

J. Martinez, A. Harwood, H. Limerick, R. Clark and O. Georgiou, "Mid-Air Haptic Algorithms for Rendering 3D Shapes," 2019 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE), 2019, pp. 1-6, https://doi.org/10.1109/HAVE.2019.8921211

J. Martinez, A. Harwood, H. Limerick, R. Clark and O. Georgiou, "Mid-Air Haptic Algorithms for Rendering 3D Shapes," 2019 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE), 2019, pp. 1-6, https://doi.org/10.1109/HAVE.2019.8921211

Abdenaceur Abdouni, Rory Clark, and Orestis Georgiou. 2019. Seeing Is Believing but Feeling Is the Truth: Visualising Mid-Air Haptics in Oil Baths and Lightboxes. In 2019 International Conference on Multimodal Interaction (ICMI '19). Association for Computing Machinery, New York, NY, USA, 504–505. https://doi.org/10.1145/3340555.3358661

Beattie, D., Clark, R., Harwood, A., Georgiou, O., Long, B.J.O. and Carter, T.A., Ultrahaptics Ltd and Ultrahaptics IP Ltd, 2020. *Mid-Air Haptic Textures*. U.S. Patent Application 16/734,479.

1 Introduction

This section will provide an overview into the different topics covered throughout this thesis, including the underlying personal interests of the research such as hand tracking, haptics, virtual reality, and HCI. Within this, the overall structure of the thesis will be covered with the three projects being briefly introduced, with insights into their purpose and intended outputs.

1.1 Thesis Topics

1.1.1 Hand Tracking

Hand tracking has long been seen as one of the most prominent "sci-fi" interaction technologies of our modern era. Simply being able to naturally manoeuvre your own hand, with an interface reacting as necessary is a dream for many. This intrigue led me, and many others, into this fascinating world, where the dream of being digital hand magicians is being slowly realised.

Devices such as the Leap Motion Controller (LMC) (Leap Motion 2013) and Oculus Quest (Oculus 2019a) have brought formerly niche hand tracking technology to the masses. Barriers to entry have been significantly lowered, enabling researchers and developers alike to implement and use high quality hand tracking, without having to produce large swathes of fundamental software stacks. This has naturally led to a significant change in the types of research being undertaken with increasing amounts of work focusing on use cases instead of simply implementations.

Research has long focused on the potential of how the hand can be used to interface with computers, with a strong trend towards gesture recognition (Zimmerman et al. 1986) and replication of positioning (Nirei et al. 1996). Only within the past few years has the focus transitioned towards real-time object interactions, in part due to advances with processing speed and the emergence of low-cost devices. The understanding of 3D movement has always been tricky, especially when real-time processing is a priority. Ensuring the alignment of virtual hands to real is critical to the experience, without it the user will feel disconnected and disorientated. While it may be possible to implement such "magic trick" theories as the rubber hand illusion in VR (IJsselsteijn et al. 2006), where you're able to visually paint over reality, it's not a viable solution for long term system interaction.

Although hand tracking may result in greater accessibility and reduce friction of entry, it is still a long way off from being widely accepted or entirely efficient compared to controllers. Many sectors and businesses are attempting to implement the technology without fully understanding both the possibilities it brings alongside the limitations. Understanding and communicating those limitations will be fundamental to progression and improvement.

1.1.2 Virtual Reality

The technology industry is starting to transition once again, just as it did back in 2016 with the consumerisation of VR headsets. VR and AR devices are relinquishing their needs for a dedicated computer, including all the processing they require directly on the device itself. This trend of increased portability brought two key issues to the forefront, the first being the need for reliable inside out tracking without the need for external trackers, and the second being the option for reducing physical extra devices such as controllers or remotes.

By removing controllers, device manufacturers can focus entirely on the main headset, reducing moving parts and in turn reducing the number of physical items the user has to hold or carry with them. User comfort is one of the key pillars of VR hardware adoption and removing the controllers may help speed up this area of development.

At the start of the boom of consumer level VR devices there were a few key defining moments that monumentally shifted different companies' directions, with one such surrounding interaction. The two key players of Oculus and Valve were experimenting with distinctly different options and methods of VR implementation. Although Oculus originally helped kicked off the original interest into the sector (Rubin 2014), it was when Valve introduced "room-scale" interactions to its prototype headset where interest beyond the screen peaked. Not only were you able to see the virtual world, but you were also able to traverse within it, directly interacting with the digital environment in ways not seen in prior consumer devices. Technologies such as the Nintendo Wii may have introduced motion controls to the masses, but the effects were limited to the 2D screen, and in most cases with basic gesture based movements or pointing actions. The enabling of full 3D movement and interaction in such an accessible and cohesive package was significant across every sector, especially when prior devices generally focused on either the visuals or the input.

As with the importance of the transition to 3D interactions, the releases of the Oculus Quest and Microsoft Hololens showed the importance of devices that could be entirely self-contained without relying on a large computer to do their processing. Both of these headsets took different approaches, with the Quest being a portable VR device targeted at the masses, while the Hololens being an AR device targeted at the business sector. Similarly though, they were able to position themselves in real-time using simultaneous localisation and mapping (SLAM) , performed entirely on the device without extra processing. Many other systems at the time were relying on physical external devices to track the headset, which thus set these uniquely apart from the competition. Not only did this mean they were self-contained in operation, but it also vastly increased the possible areas in which they could be used. As there was no need for external trackers you could turn on the device and start using it straight away, drastically cutting down on setup procedures and time. These devices both support more conventional controllers, but also support controller free hand tracking, allowing the user to both interact with the systems' interfaces as well as simulations and applications.

While the transition to self-contained devices is a logical progression for the ecosystems, newer additions of novel input sensors are fundamentally changing how we interact with these devices. A recent push by new start-ups such as Lynx (Lynx 2021) and Pimax (Pimax 2021) have been revolving around the introduction of hand tracking and eye tracking technology. This is combined with other recent sensor developments such as brain electroencephalograms (EEG) and face tracking sensors. Including these novel sensors on a device could significantly increase the experience for a user, breaking past the prevalent issues faced by lack of immersion. Certain companies within the space may struggle with user preconceptions however, especially when it comes to the privacy of the user's facial, and brain wave data.

1.1.3 Haptic Feedback

Haptic feedback is a long studied and developed field, with greater focus since the popularisation of the smart phone. Prior to this, most people used standard desktop computers which usually had a fairly generic keyboard and mouse. When smart phones came to the market, they brought with them rudimentary rotational motors to produce haptic effects for incoming calls and texts. Over the years, these have been developed, improved, and refined to be applicable in all different manners of the interface experience, with greater detail and clarity. These haptic effects play a large role in helping with the perception of cause and effect on the phone, aiding and guiding the user through their flow of actions.

It's not to say that smart phones are the only popular sector of haptic feedback, market segments such as video games and training simulations have long been applying haptics to aid in their users. While smart phones are historically known to apply haptics for the role of event notification, video games and training simulations often use them for the replication of forces or touch. Replicating the sense of force and touch can play a large part in improving the overall effectiveness of a simulation, although can be incredibly challenging when wanting to precisely and accurately convey different sensations. The differences in mechanoreceptors in the skin can significantly affect the overall feelings produced by haptic devices, even within just a few small millimetres of their application.

While optical hand tracking may introduce new levels of accessibility, it sidesteps years of work into the replication of tactile perception. The removal of physical controllers makes it harder to introduce haptic feedback, with any addition to the user or their hands reintroducing the friction just removed. Researchers and developers have been working to alleviate this issue for several years, with varying levels of success and ingenuity. Technologies such as drones and focused ultrasound have attempted to replicate the sense of touch, without requiring direct attachment or grasping by a user. Some developers have opted to entirely forego the inclusion of haptic feedback, and instead rely on pseudo haptics. Each of these methods are attempting to reproduce the effects that have been lost, with a common trait of expecting a future that may not necessitate directly touching or holding devices.

The ever present world-wide shift away from physical touch and interaction due to the COVID-19 pandemic has introduced numerous issues, especially for communal points of work and socialisation. Touchscreens and location based entertainment environments have been significantly impacted, with the necessary focus on hygiene rendering many of their previous implementations or technologies far from ideal. Many social environments are far from viable in the current climate and people are significantly more disconnected due to the effects of lockdowns and social distancing. A large shift of focus has occurred in haptic research, where the replication of touch is being attempted through technologies that do not require physical contact. Although the desire to implement these technologies is strong, the main majority of them are still in the early stages of their development.

1.1.4 3D User Interfaces

Modern trends in user interface design have shifted away from the previous use of skeuomorphic visuals, with a strong focus on minimalism. This has been coupled with large pushes by several of the key big tech companies such as Google and Microsoft releasing their own design guidelines. Boiling down the structure and complexity of these designs has led to a world where information is generally easier to digest, while instilling relatively common types of interactions found throughout different systems. As great as this is for the realms of 2D interface design, several of these principles fall apart when brought into 3D.

Many interfaces around the initial consumer boom of VR have focused around the usage of ray-cast based interactions with 2D planes and buttons. A process of simply copying and pasting common 2D elements and principles has been repeatedly used. While these methods of implementation are not inherently wrong, they still work and function as intended, they lack any significant usage of the benefits of 3D vision or interaction.

As was the case during the introduction of the smart phone, many of the guidelines and design considerations for 3D user interfaces within stereoscopic HMDs are not yet set in stone. This leads to wildly varying designs, as well as significantly differing methods of interaction. Creating largely differing interfaces can cause significant learning requirements between applications, and in turn reduce the effectiveness of the system as a whole.

Unlike with smart phones and standard computers, there are considerably more possible input methods and interactions available within the world of VR. Due to this, it will be significantly more challenging to produce all-encompassing guidelines for 3D interfaces.

1.1.5 3D Object Interaction

Simulations and training systems have long implemented 3D object interaction, be it through a controller, mouse, or secondary tool. As technology has evolved, so have the degrees of freedom available to be used, going far beyond the simple movements of the mouse. Smart phones have brought gyroscopic controls as a prominent sensor available for developers, allowing for natural methods of movement and pointing, through tilting and rotating the device. While this allows for easier movement, it rarely takes on a significantly different physical interaction, usually working as a static 2D point of interaction.

VR devices have full access to room-scale 3D interactions, allowing the user to physically adjust their hands and head directly, without having to act through proxy. These are usually performed using a physical controller, where the user will be holding a device and pressing or grasping some form of button. Although this generally makes for great interactions in this scenario, there are numerous issues still to be solved, especially when the degrees of freedom increases.

Hand tracking is the prime victim of this, where for the longest of times researchers and developers have struggled to effectively implement much more than sets of gestures. The surge of developments in the sector have resulted in devices that are now able to track both hands effectively, to the point where full 3D interactions are stable. With this, the focus once again shifts back towards the possibilities of how a user will interact with the virtual world around them, and to what interactions will they be using to do so.

1.1.6 User Study Recordings

Research studies, especially within academic scenarios, have historically sometimes treated the process of real-time data collection from users as an afterthought. The studies and technological advances are of great interest, however, there are numerous instances where the resulting findings are simply qualitative data. This is not to say that the results are not of importance, it's more that analysing a technological advancement with pure qualitative data can hide or obscure possible benefits or negatives. Ensuring consistent, beneficial data should be at the forefront of every study, yet there is often a strong desire to only present results that show improvements.

Throughout my prior studies I have attempted to implement and make use of real-time data collection where possible, developing custom methods each time. While there are other tools or frameworks available, they generally compromise on certain key factors of their implementation. Crucially these limitations have generally been the type of data recorded, the flexibility of expansion with custom code, the impact on the underlying system, and the required extra costs. Developing a fundamentally different approach to data recording, while ensuring the method of implementation creates as little friction as possible could result in a highly useful set of tools for researchers.

1.1.7 Topic Relationship

Throughout this the thesis, the above outlined topics will be interchangeably covered and analysed, due to their intrinsic linking throughout the different chapters. Crucially, several of the topics can directly impact the effectiveness of the other.

Optical hand tracking provides users with enhanced immersion and reduces barriers to entry within virtual reality applications, by directly replicating their hand movements and motions in 3D space. While it can be used within 2D screen scenarios, it produces significantly greater impact when the user is also immersed within a 3D head mounted display. Although it is showing promise as an emerging commercial technology, there are still questions surrounding its implementation, as well as the interactions that need to be undertaken by the user.

One of the key selling points for many years of games controllers has been that of rumble, or more specifically force feedback. This has continued with their importance within the mobile phone market, with haptic feedback motors being present within virtually every device currently available. They play a key role in helping with secondary information for the user, without having to directly rely on a specific effect to provide contextual information. While these are perfect for when the user is holding something, this goes directly against one of the main selling points of optical hand tracking, that of being entirely hands free. Providing this haptic information to the user could still be integral to their experience, and crucial to meeting user expectation.

By focusing on two key aspects of 3D interaction, both interfaces and objects, it allows us to explore crucial computing paradigms that users will, and do, throughout virtual reality applications. Focusing on precise user study data recording will allow us to produce greater levels of detail within our results.

1.2 Research Questions

The topics covered within this thesis can be summarised with a few research questions. Firstly, we plan to cover information related to the advancement of 3D user interfaces. We are fully aware that the scope of 3D UIs is significant, and larger than what can be covered in the projects of a thesis. Due to this, our scope will be focused on that of reactionary information and 3D buttons. With this in mind, our initial question is that of:

Q1. What is the effect on user opinion and preference when various visual modalities of a 3D buttons' reaction are modified?

Secondly, we intend to understand the greater differences between using optical hand-tracking and more conventional controllers. Many companies are starting to make use of unique controller features to try and differentiate themselves among the ever-evolving VR headset market. Several companies are trying to improve their current types of controllers, while others are attempting to use hand tracking as a first-class input. Helping to understand these differences, especially within grounded and focused contexts, will improve the chances of future implementations. To this extent our questions are that of:

Q2. What are the differences in performance and efficiency when utilising hand tracking or conventional VR controllers?
Q3. What interactions and tasks are benefitted the most when using different VR input methods?
Q4. Do the performance statistics of varying independent variables coincide with the opinions of the participants, when using different input methods?

Notably, these questions are significantly lacking in a crucial area of both the topics, and the placement companies' key selling point. With the introduction of optical hand-tracking comes the loss of many types of possible haptic feedback. Understanding whether this can be reintroduced through the usage of mid-air haptic feedback, and to what quality and effectiveness will be paramount to the success of this technology throughout VR and beyond. From this we can question:

Q5. What are the performance and efficiency effects on the user when mid-air haptic technologies are applied?
Q6. What are the effects on user opinion when mid-air haptic feedback is introduced?

With the above questions in mind, we were able to focus our research into a clearly focused direction. Each of the questions fall into one of two main categories, either being that of a quantitative question or a qualitative. While the qualitative questions can be achieved through the usage of surveys and questionnaires, the quantitative ones are more challenging. These questions needed a solution that could work beyond what a simple video feed could provide, where statistics can be reported and converted, while still performing effectively within VR.

1.3 Thesis Structure

This thesis is presented in three main parts, summarising the work of three different projects.

The first chapter consists of work surrounding 3D interfaces, exploring the effects of hand tracking and haptic feedback on the user perception of varying types of button reactions. A user study was undertaken within VR where participants would interact with the different modalities as well as answering Likert rating questions about said buttons. It focuses primarily on the differences between the buttons and the effect mid-air ultrasound haptics provide.

The second chapter consists of work on the differences between controllers and hand tracking when tasked with basic object interaction. These input devices were compared through a user study with their relevant haptic modalities of LRA haptics for controllers, and mid-air ultrasound haptics for hand tracking. Participants were tested across two different fine motor skill tasks, where the tasks would require different types of motions to complete them. Each of the participants movements, performance metrics, and overall opinions were extensively recorded and analysed.

In the third chapter I cover the analysis toolkit created and used throughout the user study of the second chapter. It goes in-depth into how the toolkit was designed, developed, and then subsequently implemented within the simulation. Comparisons between other recording systems and options are drawn, as well as a direct case study between the first and second chapters study recording methods. Unlike the prior two chapters which present the results from user studies, this chapter resulted in an open-sourced toolkit, free for other researchers to implement and use within their work.

Finally, conclusions from the work as a whole are drawn and discussed, with a look into the future work that could be undertaken for each project.

1.4 Placement Company

This thesis covers work undertaken within an engineering doctorate. The entirety of this research was performed at the, formerly Ultrahaptics now, Ultraleap offices in Bristol. Within the company there has been a significant shift towards understanding the metrics as to which hand tracking and mid-air haptics can be used to improve the interactions found within virtual reality. The insights in this thesis have been used to help improve understandings and products, while developing useful software and tools to enhance the continued approach to user research and data collection.

1.5 Accompanying Video Material

The work in this thesis can be viewed through an accompanying video. This video covers the different topics and projects presented through the thesis, including other contributions and preliminary research undertaken at Ultraleap. It can be viewed at https://youtu.be/zaxWCBxaIxg

2 Literature Review

This literature review will be covering the core topics of the thesis: interactions within virtual reality HMDs, and haptic feedback.

2.1 Head Mounted Virtual Reality Interaction

Head mounted virtual reality (VR) has received significant research and consumer focus in the past 5 years, with numerous consumer and business focused devices coming to the market. Devices such as the HTC Vive (HTC 2016a) and Oculus Rift (Oculus 2016) brought multiple years of combined research to the masses with relative ease and affordability. Not only do these devices allow for people to be visually immersed through full 6 degrees of freedom (6DOF) head-mounted displays, but they also introduce 6DOF controllers for interaction. As technology has improved over the past few years, so have the options available for VR interaction, as well as the surrounding theorems.

Interaction research can be split into two distinct groups, physical technologies for interacting, and technical methods of interacting. Methods of interaction can be classified in a number of ways, primarily by whether the interaction modifies the individual themselves, or whether it affects the simulation they are in. These can then be broken down further, with self-interactions being either movement or presence related, and simulation interactions containing object selection and manipulation, and system and interface control. Within this review we will be focusing on object and interface related interactions, as they are directly related to our research. Interface interactions are one of the core segments of system interaction, where a user will interact with varying types of menus, buttons, and proxies, to achieve their desired results. By understanding the literature this key area of interaction, we can help to ensure we produce scientifically significant studies and data.

2.1.1 Hand Interaction Devices

In the book "3D User Interfaces" p.349 (LaViola et al. 2017), physical hand-based interactions can be grouped into two popular setups. Either the hand will be classified as a whole object, which can be called a "power grip", or the hand will grasp an object which is in turn called a "precision grip".

> *"When the device is directly attached to the hand, all translation and rotation operations are carried out by larger muscle groups of the user's shoulder, elbow, and wrist. In contrast, with the precision grip, the user can use smaller and faster muscle groups in the fingers. The results of experimental studies demonstrate that a precision grip usually results in better user performance, particularly in 3D rotation tasks. [...] So, as long as the design of the device promotes using fingers for 3D manipulation, user performance benefits. Because a spherical shape is easier to rotate in the user's hand, ball-shaped devices are preferable when precise and efficient manipulation is required."* (LaViola et al. 2017).

This strongly relates to common trends seen of both commercial and research related hand-based input devices, where they generally fall into one of the following categories.

- A hand-held controller that conforms to the hands' palmar surface and follows the users full 3D position, such as a wand (HTC 2016b) or tool.
- A hand or limb encapsulating device, where it surrounds the body part, such as a glove (Zimmerman et al. 1986) or armband (Visconti et al. 2018).
- A hand-held device mounted to another surface or object through which the user can manipulate a digital representation, such as the Phantom Omni (Sensable Technologies 1994) and Novint Falcon (Novint 2006).

- A hands-free device that tracks a part of the body through imaging or other types of sensors, such as stereo IR cameras (Leap Motion 2013) or ultrasound (Ogris et al. 2005). In this scenario, ultrasound can be used to track by acting as both an emitter and receiver by listening to the reflection of the ultrasonic waves.

Many papers have implemented Leap Motion or Microsoft Kinect hand tracking as a novel input method or produce their own custom physical device. These custom devices are often trying to mimic certain physical structures or conform to the shape of the hand to improve their effectiveness. While controllers, which provide button and force feedback through their devices, are commonly found, there is an overarching desire for the improvement of hand tracking technology as the input is seen as more natural and accessible (Ens et al. 2016). Gesture recognition is well researched, however, usually tends to lead to issues with disembodiment and excessive learning requirements compared to other control methods. Context is necessary when using gestures which doesn't ensure that they can be universally adopted or implemented.

An underlying trend for research often comes down to the accessibility of the technology during development. Several papers noted that it was easier to implement the HMD specific controller when using in-hand technologies, while other commented that the Leap Motion device was the easiest hand tracking technology to implement and most reliable of available options at the time.

2.1.1.1 Hand-held Controllers

Hand-held controllers are usually trying to replicate the level of flexibility and efficiency found through specialised tools or the keyboard and mouse.

Commercial video games hardware have commonly opted to distribute alongside of a handheld controller for proprietary interactions with their device. For the longest of times these were hardwired directly to the machine such as those found on the original PlayStation and Xbox devices. As time went on, these became wireless, allowing for users to be positioned anywhere and the controllers held as they wished. These controllers were still relatively passive in their interactions, with users still simply pressing a button or pushing an analogue stick to interact. With the introduction of the Nintendo Wii in 2006 (Nintendo 2006), motion controllers were introduced to the masses, where the use of IR sensors and a 3DOF accelerometer allowed players to swing and point at items on the screen with ease. While the overall accuracy was not particularly impressive with the Wii by modern standards, it influenced a large number of future controllers. The PlayStation Move (Sony 2010) provided a similar but more refined experience to the Wii, and was subsequently used within the PlayStation VR headset (Sony 2016). Designs and functionality cues can be seen within the HTC Vive controllers (HTC 2016b), where they implement a wand based setup with relatively few physical buttons, relying heavily on their motion and triggers.

Figure 1 The Nintendo Wii (Nintendo 2006) controller on the left, and the PlayStation Move (Sony 2010) on the right. The Wii controller uses IR cameras in the device to track IR lights,

while the Move uses a camera attached to the console to track the coloured spheres on the controller.

Large amounts of research arounds controllers focuses around the development of custom physical hardware to perform an action that is currently challenging using readily available hardware. In a study by (Azmandian et al. 2016), passive haptic effects were used to enhance the VR experience of the user. These were implemented in a way where their single physical prop was modified through haptic to effects to feel like multiple different types. This improved the sense of presence compared to wand-based 3D controls, such as those in the HTC Vive. Coincidentally, the application of the haptic effects resulted in higher user satisfaction and limited the number of interaction side-effects. (Choi et al. 2018) developed a multifunctional handheld haptic controller for interacting in virtual reality. It allowed for users to grasp, touch, and use triggers from a single device, while providing haptic feedback.

This image has been redacted for copyright reasons.

Figure 2 The multifunctional controller built by (Choi et al. 2018). In this instance it's showing a virtual can being grasped and providing force feedback.

(Pham and Stuerzlinger 2019) compared the performance of multiple input methods in VR and augmented reality (AR). A pen-like interface was compared against HTC Vive controllers and a mouse for selection and pointing tasks. They found a number of key results, where the pen interface performed better than the controllers and that the pen was comparable in performance to the mouse. Limitations came into play however, as the pen would not be able to replicate the effectiveness of the buttons on the controller or mouse.

The study by (De Paolis and De Luca 2020) looked into the differences between the HTC Vive controllers and Myo gesture armband as an input device within VR. The Myo armband uses accelerometers and electromyography sensors to register movements of the arm. Their task required users to explore organs of the human body and navigate within them. Results showed improved usability of the Vive controllers, even if the required learning amount was similar. The Myo was shown to have significant negative impacts on immersion and adaptation to the virtual environment. This could be attributed to the lack of positional data provided by the Myo, simply relying on accelerometer information, or the highly novel method of input. There could be a significant amount of prerequisite learning required to truly reap the benefits of the device, something that is challenging to achieve within smaller research studies.

(Caggianese et al. 2019) compared the HTC Vive controllers against the Leap Motion device with object manipulation tasks. Their experiments found a preference towards the Vive controllers, but noted that the further developments are needed to improve usage within complex tasks.

Notably, many of these handheld controllers tend to rely on only a few key fingers for their interactions, with the other fingers being superfluous. This usually results in systems that have interactions surrounding the index finger and thumb, and simply show the other fingers with some form of visual representation, regardless of whether they are tracked or not. Many commercial devices are shown to be wireless, however this often not the case when looking towards research developed devices.

2.1.1.2 Hand and Limb Encapsulating Devices

Hand and limb encapsulating devices try to mimic and replicate exactly what the expected limb is doing, or use it as a proxy for performing predetermined actions. They can be directly replicating a digital representation of the real world limb, or be completely invisible within the simulation. Large amounts of research has been focused around the research and development of glove based devices.

(Zimmerman et al. 1986) developed the original "dataglove" back in 1986, which allowed for the monitoring and measurement of the hand through analogue flex sensors for measuring finger bends. (Maggioni 1993) developed a glove based gesture input device for more natural interaction, with multiple other gloves being developed over the years. Many pieces of research have focused around the usage of these gloves, where they can provide uniquely impressive hand tracking as they are not limited by field of view or occlusion such is the case with optical hand tracking.

This image has been redacted for copyright reasons.

Figure 3 (Zimmerman et al. 1986)'s "dataglove" and the resulting visual output shown on the computer screen.

Several pieces of recent research have focused about reducing the cost of these gloves. (Oqua et al. 2018) designed and developed a glove design that can be fully 3D printed, using Arduino boards for control. In a similar vein, (Liu et al. 2019) developed a novel glove-based interaction method, and compared it against a Leap Motion controller (LMC). This glove was tested within different VR grasping tasks, and was found to result in higher overall successful grasping rates than the LMC.

Most encapsulating devices are those of the glove variety. While there are other devices such as the Myo armband, they are generally regarded as having weaker accuracy (Visconti et al. 2018) or used as a purely gesture based approach (Pezent et al. 2019).

2.1.1.3 Mounted Devices

Mounted devices are usually designed to mimic those found in real-world machinery or operations. Most medical studies will make use of mounted devices in which they are controlling specialised grippers, representative of their real world counterpart. These devices are often relatively limited in their operation area, with a predetermined range of movements available.

Studies of the Phantom Omni and similar devices usually focus around their application of haptic qualities rather than directly for their interaction qualities. This can be seen in a study by (Vélaz et al. 2014) where the study found no significant benefits of using the device over a standard keyboard and mouse.

This is not to say that these devices are inherently bad, but it generally comes down to the context of use rather than their effectiveness. Many of the studies and devices are targeted towards medical and fixed position environments where they are effective, although this goes against the current trends of room-scale virtual reality.

2.1.1.4 Hands-free Tracking

Hands-free tracking is commonly performed through standard RGB cameras, however, can be achieved through a number of novel sensors. Unlike other technologies, this is usually highly software related, with most modern studies relying on readily available hardware such as the Leap Motion Controller (LMC) or Microsoft Kinect. Research tends to focus onto either the replication of hands as a direct interaction method, where the hand will be used to touch or grab interfaces and objects, or the recognition of hand gestures.

Early iterations of hand tracking mimic those found in modern day solutions, where they attempt to recreate every single bone in the hand. These early attempts were not able to effectively run at suitable latencies for real-time hand tracking. Research by (Rehg and Kanade 1994) led to the tracking of the human hand from grayscale images at up to 10Hz. Further work by (Nirei et al. 1996) combined two separate image feeds to reduces issues with occlusion and local minima. This resulted in digital representations that were close in value to their estimates, however, was not fast enough for real-time usage. (Lu et al. 2003) developed a method for tracking hands using a single camera input, relying on multiple different information channels such as edges, optical flow, and shading. The information is then fed to forward recursive dynamic model that tracks the motion based on the derived 3D forces applied to it. Their approach to dynamic estimation of hand shape model significantly improved tracking accuracy and robustness, but noted future work was needed in regards to signal filtering and background segmentation.

In 2013, the LMC (Leap Motion 2013) was released which summarised multiple different areas of hand tracking technology into a single device and technology stack. It allowed for real-time multi-hand tracking from stereoscopic IR cameras. This allowed a large swathe of research to be conducted into hand tracking, enabling new ways of interaction across multiple sectors without having to develop the challenging underlying technology stacks. With an update in 2016, the device significantly improved its effectiveness in VR, which crucially enabled robust multi-hand 3D interactions. While many recent technologies have managed to replicate the principle of robust camera based gesture recognition, such as Google's Mediapipe (Zhang et al. 2020), few technologies have been able to replicate the quality of the 3D interactions. While the original Microsoft Hololens (Microsoft 2016) was built entirely around the principle of using hands as the primary interaction method, the overall latency and lack of dexterity control resulted in a system that was only suitable for tracking very specific gestures and orientations. The recent updates to the Oculus Quest (Oculus 2019a) headset have enabled the usage of 3D hand tracking within a fully standalone virtual reality headset. Hand tracking is starting to reach new levels of precision once again, with incredibly robust and impressive inter-hand interactions presented in work by (Smith et al. 2020). Although the tracking is incredibly impressive, it once again falls heavily into the research implementation as it uses 124 cameras and takes anywhere from four to ten minutes to process a single frame, far beyond what is acceptable for real time interactions.

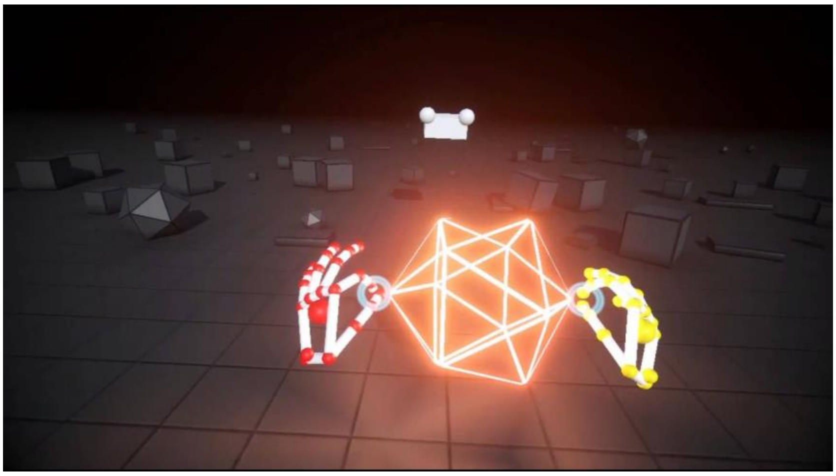


Figure 4 Image showing the default hand representations generated by the (Leap Motion 2013) device within the Blocks demo. Numerous studies have made use of these default hand models within their work.

One of the earliest popular commercial hands free interaction devices was that of the Sony EyeToy (Sony 2003). It was a custom, low power webcam that allowed for interaction with the PlayStation 2 console in a variety of games. Simple computer vision (CV) and gesture recognition allowed for several "minigames" that ran at real time, without needing any interaction from a controller. This early development is often referred to as the precursor to the original Microsoft Kinect, which was originally designed to be a video game controller replacement. The most recent iteration of the Microsoft Kinect, the Kinect DK (Microsoft 2019a), is that of a small compute unit that provides improved CV processing, both RGB and IR cameras, and multiple microphones designed primarily for business customers.

Historic preconceptions about hand tracking is that it is less accurate and precise than that of a tracked controller utilising physical tracking apparatus. To a degree, this is still the case and will remain so for a long time as issues such as per hand occlusion and finer micro movements require better algorithms and processing power than are currently available. Several studies have looked directly into the accuracy of hand tracking of Leap Motion devices. (Smeragliuolo et al. 2016) compared the hand tracking against marker based motion capture, finding it was of high quality for the hand and wrist, however, was not suitable for producing information for the forearm. This echoes prior work by (Guna et al. 2014) where the LMC was analysed, finding the controller had low standard deviation between positions, but noted significant drops in accuracy as they moved further from the device. More recent studies using more up to date software by (Valentini and Pezzuti 2017) found that fingertip tracking of the LMC was better, noting no significant differences between accuracies the further away from the device origin.

Similar accuracy based studies have revolved around the implementation of the technology in novel applications or comparisons against other devices. In a study by (Chan et al. 2015), the LMC was used as an authentication device by analysing hand geometry and gestures. They found their algorithm was able to match registered users to fresh data with an accuracy of over 98%, with an error rate of 0.8%. While not directly about interacting with objects, it shows the accuracy and finesse available with low cost, low friction devices such as the LMC. (Gunawardane and Medagedara 2017) compared hand gestures from a LMC to that of a custom data glove, for controlling a robotic soft finger. They found that the data glove produced fewer erroneous data compared to the LMC, however, the LMC data was more consistent and repeatable.

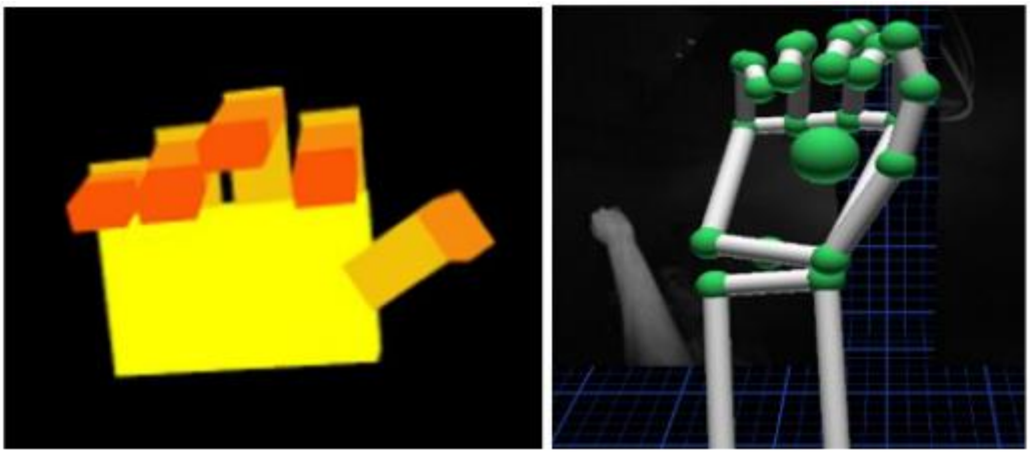


Figure 5 Picture showing the differences between the glove and LMC data from the study by (Gunawardane and Medagedara 2017) © 2017 IEEE.

2.1.2 Optical Hand Tracking Review

Throughout this thesis we will be making repeated use of optical hand tracking. While the technology overcomes several challenges, there are several cons that must be taken into consideration.

2.1.2.1 Pros of Optical Hand Tracking

When compared to many other hand "tracking" technologies, optical hand tracking provides a significant upper hand when it comes to user comfort and ease of access. One of the most important factors is that unlike glove or controller-based tech, the user does not need to physically hold or touch a device to interact with it. This significantly reduces complexity and friction when utilising it, as the user can simply use their hands freely, without having to learn where a button or analogue stick may be on a conventional controller. Coupled with this, the user does not need to attach anything to their hand. In many cases this is either: a glove, which can be challenging or cumbersome to wear, or a wired device connected for power and data, which can run the risk of tangling with complex movements, especially within virtual reality where the user is no longer visually witnessing their real hands.

Modern optical hand tracking solutions are small enough, or easily integrated onto other devices, allowing them to be worn by the user, which significantly increases their effective range. This small size, combined with the lack of physical touch means that the devices can be easily and quickly implemented into research projects or demos.

Throughout the COVID-19 pandemic, the increased importance of personal hygiene and hand cleanliness pushed a significant number of experiences out of people's desires. While VR has a bit of an identity issue at trade shows, requiring people to put heavily used devices directly onto their faces, optical hand tracking has no such problem. The lack of a physical device to touch removes the need for a user to interact with possibly dirty or infectious surfaces.

2.1.2.2 Cons of Optical Hand Tracking

Optical hand tracking is limited by a number of factors that are hard to overcome without physical changes to hardware setups. Occlusion is a huge problem for the technology, be it from a lack of cameras or simply by the user covering one hand with the other. This occlusion can result in numerous instances of dropped tracking, thus preventing the user from interacting. While extra cameras can often be introduced to help reduce the present issues, it still doesn't fully solve the problem.

Unlike many other technologies such as gloves, optical hand tracking is fundamentally based off of computer vision and machine learning techniques. These produce instances where it can be hard to compensate or fix issues within the pipeline, compared to simpler technologies that may rely on analogue forces represented through linear ranges. There can also be issues related to this such as increased processing power requirements which, if not sufficiently accounted for, can significantly reduce the performance and quality of the tracking.

Another key issue that does not affect other hand tracking technologies is that optical tracking is highly sensitive to environmental factors such as natural light and reflective surfaces. Certain visual issues can be prevented through the use of infrared light instead of visible light, such as variation in skin colour, but this still does not solve issues where the cameras may simply struggle with high contrast environments.

2.1.3 Interaction Methods

Interaction method literature will be classified into two distinct groups, active and passive interactions. Active interactions are where the body of user is directly being used to manipulate and control their desired effect, such as using a hand to grab a cube. These can also be through an object or tool, such as using a pencil to draw a line. Passive interactions are where the user is performing an action to activate a predetermined interaction, such as making a thumbs up gesture to accept, or interacting at considerable range from their target, such as pointing through a ray-cast. Active interactions are usually highly contextualised, requiring an exact position or adjustment to successfully action them. Passive interactions often do not require an exact context for them to be achieved, being indirect in their actions. Many systems will rely on the use of passive gesture based interactions to access core functions of a system, regardless of when or where the input method currently resides.

A review by (Argelaguet and Andujar 2013) compared a large number of both active and passive interaction techniques. They noted that a lack of de-facto datasets for testing, coupled with numerous different and custom hardware setups made it difficult to fairly compare research, but did note several key findings. Many of the techniques were limited by hardware factors, such as visual occlusion and input mechanism, with a general trend that most of the interactions were more physically demanding in the virtual world than the real. Optically tracked interactions struggled with confirmation of actions considerably more than any physical controller. They were unsure about what interaction will remain as a popular or preferred method of 3D manipulation, and to an extent that opinion can still be seen today. Different systems implement varying techniques for manipulating objects with no universally accepted method, just as you would find within different conventional video games and software packages. While this may make sense in certain specialist software, it introduces barriers to entry or widespread adoption that still needs to be addressed

A more recent review by (Jankowski and Hachet 2015) found that 3D interactions that provide high levels of control generally require large amounts of expertise to work effectively. Low levels of control were effective for novice users, admittedly with reduced interactivity. One of the largest challenges they outlined is the process of increasing interactivity levels, without directly increasing the required learning effect or complexity. They noted that "sketch-based" approaches, where the user would input one action and the system would effectively fill in the gaps of the interaction, were good examples of this and could help reduce barriers to entry. Another key issue is that of ensuring the interaction can suitably adapt to the increasing requirements of a task, where oversimplification of the interaction and resulting outcome can result in frustration. Notably, they embrace the fact that interfaces may not be entirely obvious or faster than current real-world or 2D interaction. They instead favour the principle of ensuring enjoyment and appeal of an interface, as well as the standard criteria of speed, efficiency, and enjoyment, are taken into account.

Beyond prior reviews, (Mendes et al. 2019) surveyed the different options for 3D object manipulation. They explicitly extract mid-air and touch based interactions as individual topics. Nearly all of their reviewed techniques for touch-based interactions had low degrees of freedom or resorted to similar effects of a mouse, with high DOF interactions usually resorting to widgets. They found most approaches to manipulate objects with mid-air interfaces rely on principles found within the real world, without any separation of transformations. Many studies have realised that the accuracy of the human hand is limited, thus necessitating helper functions. This inaccuracy can often be related back to the lack of replication of real-world forces or minute amounts of perceived latency. Improving accuracy generally occurred through reducing hand motions or by moving the view closer, which isn't entirely feasible in VR. Widget based interactions have shown poor results for hands free interactions, when compared to direct manipulations. They found that although techniques that are found to improve accuracy are effective in doing so, they can detrimentally affect the speed of the user.

While there are many different types of interaction methods available, for the research in this thesis I have chosen to group them into two distinct groupings that reflect common trends in virtual and augmented realities. Active interactions summarise types of interactions where the user is the direct vessel for interaction, be it the hand grabbing an object or directly actioning the press of a virtual button. Passive interactions provide a secondary item to implement the interaction for the user, this can be anything from using a ray to point an object, or requiring the user to perform an action that causes an effect.

2.1.3.1 Active Interactions

A classic interaction method is that of the "Go-Go" technique (Poupyrev et al. 1996), where a virtual hand representation will dynamically translate at a larger ratio than physically possible. This allows for users to reach considerably further than if they were simply moving their hand to the desired position, without having to physically walk towards an object. It is shown to be easily understood, however, in selection tasks it was less effective than simply casting a ray due to the increased degrees of freedom. Several systems have tried implementing this approach, but it has rarely been used in situations where the user has full 6DOF movement or can freely teleport. Similar types of interactions where the user will be presented with a ray-cast style of range extension are often used to prevent perception issues with disembodiment.

The replication of real world interactions has long been a topic of focus for many. (Borst and Indugula 2005) produced a physics based approach to interactions, where each finger of a hand tracked model were made up of rigid-body joints. These joints were then used to produce physically accurate interactions when combined with their custom system of virtual torsion, linear spring-dampers, and virtual friction. A separate, invisible physics hand was combined with the visual one to eliminate possible issues that may occur with object penetration. While this successfully achieved a large range of interactions, the system was noted to be "sticky", with more recent work by (Prachyabrued and Borst 2012) reducing the effect of this stickiness. The stickiness relates to where objects are likely to remain attached to the hand, when the desired outcome is the object being released or dropped.

Similar in vein to the rigid-body approach, (Jacobs and Froehlich 2011) developed a soft-body method approach of interaction. It directly relied on the perceived pressure on the object by calculating how much of the user's virtual finger profile was in contact with it. This in turn created a system where the users "force", the point beyond object penetration, on the object would in turn create higher amounts of friction. This implementation built upon the prior work of different grasping techniques as this can be implemented in both soft and rigid-body physics simulations. (Talvas et al. 2013) implemented a similar technique to this, through the usage of "god fingers". Each individual finger would have an additional contact plane that react and bend around objects on contact. This results in a method where the finger can wrap around any object mesh naturally, while still applying force. It was shown to allow for more complex manipulation of models, however, as it did not simulate soft-body physics, it did not resolve the issue of object penetration.

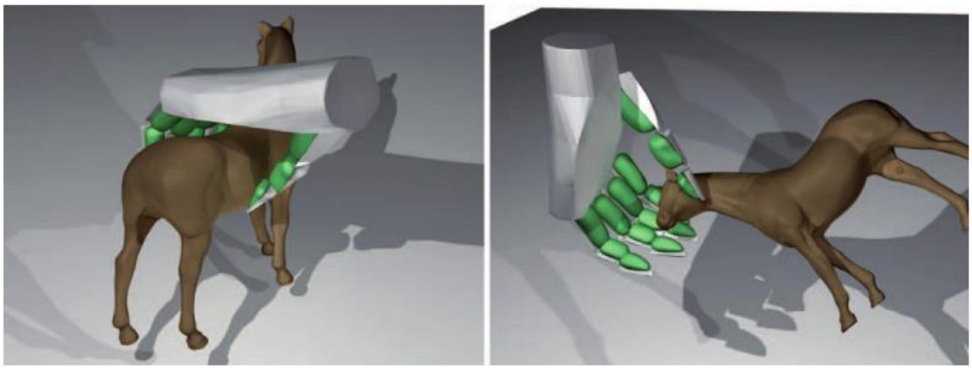


Figure 6 Image showing the grasping mechanics when soft-body physics were applied to the different joints of the hand within the study by (Jacobs and Froehlich 2011) © 2011 IEEE.

(Lin et al. 2016) produced a virtual reality platform for dynamic interactions between the user and the scene within the Unreal Engine 4 game engine. It combined the Oculus Rift headset with a Kinect, Leap Motion, and dance pad for representation, interaction, and movement respectively. It was designed to heavily mimic the real world, attempting to provide a fully body experience, with the Kinect being used for body tracking, and the LMC for fine hand movements. They noted future work to prevent objects from being penetrated by the participants limbs. While the interactions themselves were impressive, newer technologies reduce the need for several of these devices.

Many of the aforementioned interaction techniques were implemented into the Leap Motion "Interaction Engine" toolkit (Leap Motion 2017). This toolkit allows for real-time object interactions in both VR (active) and desktop (passive) modes. It implements a similar physics based interaction to the ones described by (Prachyabrued and Borst 2012) and (Talvas et al. 2013) by using an invisible physics based hand that calculates the distances between object and finger joints to grasp objects. Although it is primarily designed for hand tracking, the toolkit also supports standard VR controllers such as the HTC Vive and Oculus Rift. Similar toolkits have since been developed such as the Microsoft Mixed Reality toolkit (Microsoft 2019b) which provide slightly different features.

(Moehring and Froehlich 2011) conducted a study where they compared controllers, a flight stick, against finger tracking, which specifically detected pinches, for different types of direct and indirect interactions. Results strongly confirmed their expectations of indirect interactions being significantly easier due to simply requiring a point and button press, especially when it came to grasping and releasing. They noted that although they may have performed better, the indirect interactions were not realistic compared to the direct. Overall opinions reported good grasping and releasing capabilities for the flight stick compared to the finger tracking, but lowered judgment of grasp.

While not directly within HMD VR, (Gallo 2013) studied the effect of degree of freedom in 3D touchless interaction, with the Microsoft Kinect compared against a standard computer mouse. Two Kinect devices were used to sense hand positions, which were then converted to different transformation techniques such as translation and scale. Their results suggest that devices that allow for multiple degrees of freedom control have the potential to outperform interfaces that segment them.

(Alzayat et al. 2019) compared the differences between embodiment for different virtual reality tools as possible input methods. They compared representations of a virtual hand, VR controller, and physical tool, using a "Locus of Attention Index" (LAI) to quantify differences in embodiment. Their perception of embodiment was over a scale where the greater the tool was embodied, the more the person's attention will be on the task instead of the tool. Results indicated engagement was higher using the controller than hands, but hands were better than the tool. In a similar style of study by (Linkenauger et al. 2013), the effect of size of hand on perceived object size was questioned. They discovered that the wider the hand was, the longer the participant was likely to perceive the object, even if sizes of object had not changed.

This image has been redacted for copyright reasons.

Figure 7 The three different types of interaction used within the study by (Alzayat et al. 2019).

Active interactions have historically been significantly challenging for the technologies to perform, as the latency involved has usually been too high for them to work effectively. Consumer devices such as the Razer Hydra predate many of the advancements that were released in with the large push for VR devices in 2016. Even devices such as the Microsoft Hololens, which was released in 2016, had rudimentary hand tracking that did successfully track the hand, but with significant latency and poor finger segmentation. Many older technologies would have to pick between either accuracy or speed, however newer offerings have helped reduce these issues significantly.

2.1.3.2 Passive Interactions

Many modern VR systems make heavy use of ray-cast techniques, with ecosystems such as SteamVR and Oculus using these for their menu and interface interactions, and toolkits such as the MRTK (Microsoft 2019b) using them for a hybrid approach to object interaction. In the case of interface interactions, these are generally a simple ray that directly and linearly extends from a controller or hand and is then actioned through generic button or gesture. Object interaction is a more varied field, where systems are often seen to be adding additional helper functions to the rays. For example, the MRTK warps and bends the ray to snap towards an object within close vicinity.

Crucially ray-cast based interactions tend to be most favourable when they are instantaneous to react and linked to a hand based-origin. A study by (Nukarinen et al. 2018) found that the rays with time based hover interactions were less favourable than instantaneous button based interactions, while hand origins were favoured over head. This builds upon prior research by (Cournia et al. 2003) where gaze based interactions were shown to be slower than hand based ones.

(Pfeuffer et al. 2017) developed and tested an interaction technique that combined eye tracking and hand tracking, creating a ray-casting system where the user would look at an object within VR and then pinch using their two hands to interact and manipulate it. Their initial user feedback reported that the combined approach was intuitive to use, almost feeling magical compared to real world interactions.

Lots of research has been performed into the usage of hand based gestures for interaction within VR. (Zimmerman et al. 1986) developed the original "dataglove" back in 1986, which allowed for the monitoring and measurement of the hand through analogue flex sensors for measuring finger bends. (Maggioni 1993) developed a glove based gesture input device for more natural interaction, with multiple other gloves being developed over the years. (Khundam 2015) developed a method of first person movement in VR using palm normal and hand gestures with an optical hand tracker. They found that participants were able to complete movement tasks faster using the gesture interface over standard controllers, albeit with slightly higher standard deviation.

2.2 Haptic Feedback

Haptic feedback plays a core role in replicating the sense of touch and other forces we experience in the real world, reproducing effects that we would usually be feeling when interacting with physical objects and materials. They directly aim to provide real-time information beyond what we normally see and hear from our digital devices, without inherently making themselves directly present. These effects have become increasingly more common over the past decades, as the transition towards mobile computing has often been directly coupled with vibration motors to provide information, without having to directly look at or listen to a device. Critically, haptic feedback will continue to be a necessary field of research as (Akay 1998) pointed out, the lack of it within VR prevents full immersion due to not being able to activate one of our five central senses. Within this review we will be focusing primarily on haptic devices that target or applicable to the hand, rather than larger limbs and areas of the body.

2.2.1 Haptic Technologies

There are several common methods of producing haptics, with different technologies providing their own implementations of producing these effects. While there are many different types of haptic technologies available both commercially and in the research world, we will be focusing on those that are currently commercially implemented.

2.2.1.1 Vibration

The most common type of haptic technology is that of vibration, where the device will produce high frequency motions to generate small movements of an element within itself. Vibration haptics can be implemented through multiple different types of devices. The most common type of vibration motor is that of an eccentric rotating mass (ERM) actuator, where an unbalanced weight will rotate at high speed, powered by a motor. This rotation of the weight will cause a shaking motion due to the constant change in centroid (Precision 2015). ERMs have been commonly found in devices such as the PlayStation DUALSHOCK controllers (Sony 1997), with smaller versions being present in older smartphones.

This image has been redacted for copyright reasons.

Figure 8 Diagram showing an exploded view of an ERM actuator. (Precision 2015)

Linear resonant actuators (LRA) are a more modern and relatively common approach to vibration haptics, that produces an oscillating force across one axis instead of a rotational force. This is achieved through the use of a spring and voice coil surrounding a magnet, where the incoming voltage to the voice coil is driven at the same frequency as the resonant frequency of the spring. Overall force and power of the haptic effect can be controlled by modulating the AC input to the voice coil, meaning the higher the power input to the device, the stronger the haptic effect (Balu 2016). LRAs are more power efficient than ERMs as they do not need to physically move as many parts, while being able to maintain latent haptic effects due to the continuous storage of kinetic energy thanks to the magnet keeping the coil in place. LRAs can start producing haptics after around 10ms, considerably faster than an ERM which can only reach around 20-50ms. Common devices such as the Nintendo Switch (Nintendo 2017) and Apple Watch (Apple 2015) utilise LRAs to produce their haptic effects. In the case of the Switch, multiple segments are implemented to produce more precise haptics over a broader area.

This image has been redacted for copyright reasons.

Figure 9 Diagram showing an exploded view of an LRA actuator. (Balu 2016)

Current work and research in vibration haptics is moving towards the use of piezoelectric haptic actuators. These utilise a method of producing haptics by applying an electric signal to a material which then in turn squeezes or stretches. Unlike the prior two methods of vibration haptics, these work by placing two planes of material in a cantilever beam configuration, which then bends when the voltage is applied. Bending these materials produces motion, and is in turn repeated at high speed to generate the haptic effect. As with the transition from ERM to LRA, piezo actuators produce haptic effects at a considerably faster speed in the order of 1ms. Comparatively, the piezo actuators can modulate both their amplitude and frequency independently, which allows for more complex haptic effects (Motola-Barnes 2018).

This image has been redacted for copyright reasons.

Figure 10 Mechanical diagram showing a 2-layer bending piezo actuator. (Motola-Barnes 2018)

Vibration motors require direct attachment to the user's skin or by being held or worn by the user, but are usually relatively small and lightweight. This allows for easy implementation into devices, without putting significant strain on the user. VR and games controllers and mobile phones are prime examples of good implementations, where the devices are light weight while the implementation of haptics positively benefits the user. These devices tend to occlude parts of the hand which can be detrimental when relying on optical hand tracking solutions.

2.2.1.2 Force Feedback

Force feedback is an effect produced when a device manipulates its own movement against that of the person interacting with it (Christ and Wernli 2014). For example, if a user was pushing a virtual box within a simulation, the feedback device would provide resistance force to emulate the effect of the weight and friction of the box. Unlike vibration haptics, these devices need a relatively large physical presence as they need to physically move their housings, rather than a small actuator inside a fixed housing. A key defining trait of these types of feedback is that they are generally directly coupled with an input modality. This means that whatever device is currently being used to interact with the system will be the direct point of output for the force feedback.

Common implementations of this technology can be found within video game steering wheels, where the wheel will attempt to rotate in the opposite direction to the user. Commercial devices such as the Phantom Omni (Sensable Technologies 1994) and Novint Falcon (Novint 2006) provide 3D force feedback through the use of an adjustable arm (Omni) or three adjusting arms (Falcon). More recent devices such as the latest iteration of the Sony DUALSHOCK included with the PlayStation 5 (Sony 2020, p. 5) introduced triggers within the controller that provide force feedback.

Devices such as the Dexmo Haptic Gloves (Dexta Robotics 2021) are directly attached to the users hands and physically prevent the user from closing their hands, emulating forces of gripping and grabbing objects. These allow the user to still maintain freedom of movement, especially within room-scale virtual reality (Wikipedia 2021a), while still producing feedback.

Most force feedback devices need to be placed or affixed to another object such as a table or stand due to their size and force produced. Many medical and professional business training devices make use of this technology, with the devices usually taking up the largest physical footprint of haptic technologies.

2.2.1.3 Ultrasound

Ultrasound haptic feedback is an effect produced through the usage of focused ultrasound beams which create highly localised points of pressure, called focal points (Iwamoto et al. 2008) (Carter et al. 2013a). While most haptic devices directly require the user to be in contact with the device, producing haptic effects by modulating their housing or an element within it, ultrasound haptics take a different approach. The method of application comes from the change in pressure in the air around the user's skin, directly displacing the skin. Multiple focal points can be used, along with multiple methods of adjustment to produce the haptic effects. Devices that produce ultrasound haptics are incredibly niche, with Ultraleap producing the commercial Ultrahaptics STRATOS Explore (Ultrahaptics 2018), and research projects stemming from the Shinoda group in the University of Tokyo (Hoshi et al. 2010).

Devices producing these effects utilise multiple ultrasound transducers, where each transducer has its phase and intensity individually controlled at rates of up to 40kHz. The fast update rates and finite control of the transducers allows for incredibly rapid adjustments and repositioning of the focal point. When compared to the other technologies, ultrasound haptics provide a key benefit where the positioning of the haptics is entirely freeform within a given area, without requiring contact or attachment to the user. This makes it considerably easier and faster to produce haptics on multiple different people, without having to adjust setups. Varying layouts of transducers can be used such as square or Fibonacci spirals (Price and Long 2018).

Like the other haptic technologies, the skin will only feel the effect of the ultrasound on the displacement and release of pressure on the skin, requiring some form of modulation of the focal point. Modulating a focal point can be done in a number of ways, with the two main types being amplitude modulation (AM) and spatiotemporal modulation (STM) (Kappus and Long 2018). AM works by simply positioning a focal point at a desired 3D location, and then changing the amplitude from the desired amount and back to zero repeatedly. Multiple focal points can be used in tandem to produce AM effects at different areas on the palm. STM works by moving a focal point at high speed in repeated paths, producing the displacement by physically moving the point of pressure across the path. Unlike AM effects, STM tend to be more efficient as they only requires the usage of a single focal point to produce larger areas of effect.

A major caveat of ultrasound based haptics is the overall lack of force they produce, where they can only be feasibly felt on a person's palmar surface. Due to the lack of definition in the mechanoreceptors of the skin beyond the palms, the effects are strongly diminished even when applied to just the back of the hand. Most ultrasound arrays also use considerably many more individual components than other haptic devices, as they require a large amount of ultrasound to produce enough force to displace the skin. This increase in components directly translates into a greater amount of required power to create the haptic effects.

Ultrasound haptics devices are generally slightly smaller than force feedback devices, but still require being attached to another device for their placement. They do not necessitate being attached to the user which makes them considerably easier to combine with optical hand tracking.

2.2.2 Reviewing Haptics

Multiple varying studies have found positive benefits from the application of haptics across different disciplines and sectors.

(Teather et al. 2010) conducted research into the effect of passive haptic feedback by comparing against the standardised pointing and tapping task outlined in ISO 9241-9. Haptics were applied through a plastic panel to which the participant touched with one hand, and pointed using the other hand. They observed that pointing throughput was significantly higher with haptics than without, but did not find significant differences for accuracy or speed individually.

Within simulations, haptics have long played a role in aiding participants, both in learning and rehabilitation. (Han and Black 2011) developed a multimodal simulation where elementary students would learn how gears work, using haptic joysticks. Three different haptic conditions were applied to different students, force and kinaesthetic, kinaesthetic, and non-haptic. Students who received either haptic condition retained and recalled information better than those who received no haptics. Interestingly, the use of pure kinaesthetic haptics did not provide a substantial difference compared the lack of haptics, indicating a need for force feedback as well.

In a review by (van der Meijden and Schijven 2009), there was no decisive consensus about the importance of haptics in robot-assisted minimal invasive surgery simulations, however, the majority of studies showed *"positive assessment of the benefits of force feedback ... [and within] VR training, results indicate that haptic feedback is important during the early phase of psychomotor skill acquisition"*. This was backed up in a study where (Zhou et al. 2012) produced a VR surgical simulator for laparoscopic skill acquisition where haptics conditions were directly compared against one another. Haptic effects were shown to be beneficial for a laparoscopic suturing and knot-tying task, although crucially this was only the case during the first 5 hours of training.

(Våpenstad et al. 2013) tested the difference between haptic and non-haptic feedback within laparoscopic simulations. They found that their haptic implementation of friction was unrealistic, with 95% of participants believing it to be too strong. While 85% of the participants thought that the haptic feedback attempted to simulate the resistance in tissue, only 10% of those thought it succeeded in doing so. This result appears to be due to mismatch between expectations and realised effects. Våpenstad produced a follow up study where a criterion-based training program of a simulator with haptic feedback, produced through a Xitact IHP device (Våpenstad et al. 2017). The control group without haptics were found to produce better results than the group with haptics. Both of these studies were focused around indirect, tool based interactions where they were attempting to recreate precise, real-world forces.

Hagelsteen et al. found significantly better performance, through reduced error, from trainees when haptics were applied in their laparoscopic VR simulator (Hagelsteen et al. 2019). While the resulting haptic effect, produced through joysticks, had limited fidelity when compared to the real world, the participants still benefited from the application of haptics.

Francone et al. found improved efficiency and overall safety of patients during their preretinal membrane peeling simulation (Francone et al. 2019), which utilised CHAI3D (Conti et al. 2003) and a custom surgical haptic cockpit.

(Nordvall 2014) produced a haptic game interface, where the participants would utilise an Xbox game controller to play a game of pong, with and without haptics. Results showed that haptics provided a significant effect on the user experience, more so than visual and auditory effects.

In an attempt to improve the typing experience on portable flat keyboards, Ma et al. applied haptic and auditory effects to the typing experience of a flat keyboard (Ma et al. 2015). They found that the application of haptics increased the overall typing speed of the user, and discovered no major difference between the type of haptic effect being applied. It was also observed that only applying auditory feedback was not as effective as when in tandem with haptic feedback. A key point was witnessed where participants preferred the application of haptics on a per typing finger level, instead of the keyboard as a whole.

(Wu et al. 2017) developed a virtual reality keyboard which utilised a hand-tracking exoskeleton, P5 data glove, and micro-speakers to produce haptic effects. The virtual keyboard was compared with and without haptics, and against a conventional real-world keyboard. No significant differences were noted between the haptic or non-haptic virtual keyboard. Notably, participants were still faster at typing using the conventional keyboard even in situations where they had to physically remove the headset to see the keyboard.

(Kreimeier et al. 2019) evaluated different types of haptic feedback on the presence and performance of manual tasks in VR. Participants would be asked to throw objects at a target board, stack blocks, and identify information on blocks. They implemented two different types of haptic devices, a custom vibrotactile glove, and a Sense Glove DK1 (SenseGlove 2019) for providing force feedback. Vibrotactile feedback was reported as having the best sense of presence over force or visual feedback. Force feedback lowered throwing times, but reported issues in general with replicating the release of a ball in VR. This applied to their stacking task too, where force feedback was generally faster. Force feedback was the most preferred by participants for object throwing, while vibrotactile was favoured for block stacking. Users performed worse during the identification task with force feedback, however, this was reported as possibly being due to technical limitations.

(Kim, Jeon, et al. 2017) developed a low cost hand oriented haptic system for use with a Leap Motion controller and Arduino-based sensors. Haptic emitters were attached to the thumb and index finger tips, controlled through a wristband. These emitters produced both vibration as well as heat through the use of resistors. Immersion and presence were tested, with participants reporting stronger immersion when haptic and heat effects were applied.

(Park et al. 2011) compared the differences between haptic effects and button clicks on touch screens. They found that short duration effects for button clicks were of great significance, as had been the case in prior work (Koskinen et al. 2008). Significant benefits were found from using an LRA actuator compared to an updated form of it called a dual-mode actuator (DMA).

(Son and Park 2018) applied haptics to the palm and fingers to provide improved tactile perception of large objects, through the usage of custom exoskeleton hand-based haptic device. They discovered that the haptic application to the palm and fingers provided better perception than with just feedback to the fingers. This style of research was continued where (Park et al. 2019) studied the effect of haptics on the perceived size of virtual objects. Multiple haptic effects were compared, with force feedback and cutaneous feedback on the fingers combined with skin-stretch feedback or vibrotactile feedback on the dorsum of the hand. Haptics were applied at the moment of contact with the virtual object. A preliminary experiment found cutaneous feedback did not directly produce a significant effect on the perceived size, however, led the participant to grasp using less force compared to force feedback. During the main experiment it was found that participants would perceive objects with a size of 40mm larger with skin-stretch haptics than without, and objects of 20mm and 40mm were perceived smaller with vibrotactile feedback.

(Gaffary and Lécuyer 2018) reviewed the application of haptics within a car improves the safety of driving. Through their review, it appears that haptics were effective in reducing the visual workload, while useful for conveying information such as hazard prevention. Most implementations are designed for two key areas, first to aid the driver in their desired actions, such as interface navigation, and are commonly applied through the dashboard and steering wheel. Second, for the reiteration of warning signals and general safety of the driver, designed to increase awareness. They noted that many of the studies were performed under simulation, meaning they do not take into account real world stress or may not account for overconfidence present within simulations.

(Fröhner et al. 2019) researched the effect of wearable haptic devices for improving embodiment of virtual limbs. They compared three different haptic conditions of vibrotactile feedback, force feedback, and no feedback. Findings showed haptic feedback significantly improving the subjective embodiment of the limbs, while force feedback led to stronger perceived ownership of the limb.

Haptic technologies have been compared against one another over the years as varying advances have been made within the sector. When comparing technologies and effects, just noticeable difference (JND) studies are commonly used. This is the implementation of the Weber-Fechner Law where "a measure of the minimum difference between two stimuli which are necessary in order for the difference to be distinguishable". Using these types of study designs allows for deeper analysis as the cutaneous and kinaesthetic difference vary significantly between users, while they also produce quantifiable differences between haptic effects. While this is applicable for comparing the output of the technology, it does not take into account the logistical effectiveness or compatibility with other technologies.

These haptic technologies can be grouped into three categories: wearable, held, and contactless haptics. Wearable technologies require the device to be either attached directly to the skin or body part of the user to produce their effects. Held devices simply need to be held within the hand of the user during use. Contactless haptics require no direct attachment or prior motion of the user to feel the haptic affects, targeting the user directly through the air.

2.2.2.1 Psychology of Haptics

The effectiveness of a haptic device is a significantly different metric compared to that of visual or auditory devices. Measurements are generally harder to standardise, and often heavily rely on the usage of opinions and preferences.

In a review by (Hadi and Valenzuela 2020) across four different studies, they found that the addition of haptic alerts to accompanying message based information provided beneficial performance gains to the related tasks. These were shown to be driven by an increased sense of social presence and engagement, also shown as a heightened exchange between the user and the technology.

A rather interesting combination of topic and application was undertaken by (Ahn et al. 2019) where the calorific content was used to represent the weight of different foods. The study provided the user with two scenarios, one having objects of the correct weight, and the other with the weight being relative to the number of calories the food item contained. From this, it led to greater spatial presence and contributed towards a shift to recommended healthy behaviours.

Another study by (Webb et al. 2022) looked into the usage of haptic technologies within a virtual reality learning environment for learning. While the inclusion of the VR environment and collaborative approach significantly improved the results of the students, the addition of haptic feedback did not directly contribute to any significant improvements.

2.2.2.2 Pseudo Haptics

While most studies about haptics are working to introduce novel ways of producing haptic feedback through custom hardware, pseudo haptics buck the trend by using other modalities or the user themselves to emulate haptic feedback. Our senses can be tricked into adjusting our perception of a force when we combine one effect with another. The McGurk effect (Mcgurk and Macdonald 1976) is a prime example of this in action, where a person can have their sense of hearing confused by the introduction of a different visual stimuli.

As with regular haptics, pseudo haptics can be applied to different tasks and interactions with varying levels of success. A study by (Taima et al. 2014) introduced pseudo haptics to their interactions with lifting objects to emulate different weights. They implemented this by increasing the visual translation of an object when it was lifted, meaning that if a user moved an object 5cm up, it would visually rise by a greater factor such as 10cm. They found that by changing this visual response, it reduced the overall fatigue of the user. This line of work was continued by (Samad et al. 2019) where they were able to effectively change the perceived weight of objects by simply reducing the overall distance of movement.

Pseudo haptics can also be used to supplement applied haptics, such as in work by (Hachisu et al. 2011) where they applied visual effects on top of vibro-tactile haptics to enhance their effectiveness. They can also be used to adjust our perception of the haptic effects, where work by (Peeva et al. 2004) found the adjustment of pitch of the audible sound had a large effect on the perceived roughness.

2.2.2.3 Summary

Through the literature I discerned that most haptic technologies and research can be grouped into three device categories and two perception categories. The haptic device categories are wearable, held, and contactless, while the perceptual are realism, and informative. These were formed from the numerous studies that either try to produce or refine new types of technologies, coupled with studies that would then try to implement and iterate on the software and techniques that these technologies could output.

Worn and held haptic technologies are usually able to provide both the realistic and informative type of haptic feedback, while contactless technologies are usually only able to produce informative feedback. This is often due to the fact that the contactless technologies are not able to produce the finer required stimuli to create these realistic effects.

Haptic technologies that are worn are usually focused around the hand as this has greater definition of mechanoreceptors in the skin. They often combine haptic forces and tracking technologies into one device, but are often wired to a computer directly due to the number of different sensors and actuators involved. The most common type of actuators used throughout the different studies are simple or cheap vibration actuators that are placed at multiple different points on the device to stimulate different mechanoreceptors or parts of the body. Interestingly, several pieces of research have focused around the application of devices that are not on the hand, such as wrists or feet, allowing the user to retain full control with their hands.

Held haptic devices are usually less about the perception of textures or vibrations and usually focused towards the creation and emulation of forces. This results in devices that are larger than most or are entirely tailored to a very specific set of use cases, such as rowing a boat or swinging a sword. As with worn devices, these are usually wired and take up the entirety of the hand making it challenging to integrate with other technologies.

Contactless haptic technologies do not restrict the user's hand movements, but do generally restrict the area of application. This is inverse to the other two types, where the prior two are restricting what can be applied to the person. Uniquely, most contactless haptics require no prior application to the user, and tend to track the point of haptic application through a secondary sensor. This allows for greater levels of real-time customisation, but can lead to issues regarding the overall accuracy of application. Optical hand-tracking often works well with contactless haptics as it is one of the few types of haptic feedback without the need to occlude parts of the hand, thus improving the chances of being successfully and accurately tracked.

As a whole, the production of realistic haptics can be incredibly challenging. Many attempts to do so result in either an overly heavy approach that exceeds their original reference, or produces something that does align with the intended output. This can end up being more detrimental than positive, especially within training simulations as it can result in either significant distractions or teaches the incorrect effect. Studies that are developing or researching realistic types of haptic feedback are often solely focused on said topic, as it generally requires a significant amount of work.

Informative haptics are usually easier to implement and study as they can be repeatable and relatively contextless. This allows for haptic effects to be taken from study A and implemented into study B without significant changes or differences. As the effect is not trying to emulate something that we naturally feel, the expectations of the output will appear less distracting or confusing to the user. Ideally, they will result in almost subconscious acceptance of the feedback to the point the user treats it like any other visual or auditory reaction that they can rely on for information.

2.3 Literature Findings

The literature review shows a significant portion of research into haptic devices, but less work towards their implementations and use cases. VR appears to be in a flux between cutting edge research, and serious implementations of hardware and software. This is similar to the technological situation that this thesis's work sits, where the devices are both established (optical hand tracking) but also highly futuristic (mid-air ultrasound haptics).

With this in mind, there is a significant gap in the overlap between VR, optical hand-tracking, and mid-air haptics. The individual pairings of VR and optical hand-tracking, and hand-tracking and mid-air haptics have received a large amount of research, however, the trifecta has been shown to be less realised. This could be partly attributed to the complexity of using the different technologies together successfully. For example, most of the studies using VR and hand tracking rely on a tracking device attached to the headset, while most mid-air haptic studies use a hand tracker attached to the desk or haptic device. Combining multiple camera perspectives, while retaining high accuracy could be a significant issue that needs to be overcome.

Many of the prior studies tend to focus more towards the developments of the technologies, rather than their implementation or direct comparisons with prior technologies. This led us to the question of whether these technologies are ready for broad consumer usage, or how can we test to see whether they are. Crucially, two key areas of user interfaces and object interactions are of major interest.

User interfaces within VR have stagnated for a few years, with many implementations simply relying on laser pointer style interfaces on 2D planes. Answering the question as to why these are the favourites, compared to 3D Interfaces is something beyond the scope of the work here. We can however, research and understand part of the interface pie, by looking at some of the core underlying elements of UIs.

From these findings, we can position this thesis for impact by combining these technologies with VR and haptic wide market issues. These questions will encompass common issues with user interface design, coupled with differences between input devices and object interactions.

3 Comparing 3D Button Behaviours and the Role of Haptic Feedback within Virtual Reality

3.1 Introduction

3D user interfaces have been well researched over the years, exploring many forms of input and output mechanisms. The convergence of contactless technologies and virtual reality is becoming increasingly important, especially within the current socio-economic climate, developing for this will bring challenges many seasoned interfaces designers may not have encountered. While interfaces have shifted to newer realms, many have resisted making use of the extra depth that this often brings, due to inexperience with the technology, or quite possibly lack of research into the field. Implementing these new technologies brings numerous new possibilities, while also presenting the opportunity of iterating upon older proven mechanics.

Although the designs and contexts may have changed over time, the button has remained at the core of almost every user interface, across decades of work (Robertson et al. 1991). The overall process of pressing and actioning an item through a button is so incredibly fundamental to the usage of any current 2D digital interface, that it is entirely expected the transition towards 3D will strongly echo these prior principles. Not only will these new interfaces require research and development into their best practices, but as novel technologies become the norm there will be further challenges to overcome.

Advancements in hand tracking technologies have led to greater opportunities for natural interactions, allowing the user to simply move and interact with their hand without having to physically hold or grasp a controller. This has allowed for heightened immersion and greater accessibility, especially within the realms of VR. With the relinquishing of controllers comes a number of challenges, but primarily the removal of all tactile feedback that would've been produced through buttons and actuators. Novel devices such as focused ultrasound haptic arrays provide mid-air haptic feedback without requiring the user to physically attach or hold a controller. These devices can be combined with hand tracking to help replicate the missing haptic effects, while still retaining the flexibility of hands-free input. Prior research into interactions with user interfaces or buttons with similar technology has generally focused around the positioning and spacing of buttons (Park et al. 2020), gestures (Nor'a and Ismail 2019) and keyboard replication (Kharoub et al. 2019), and more generalised "buttons" (Hwang et al. 2017).

Within this paper we will provide the interface designer with a set of useful, and well-defined guidelines into how they can effectively enhance the button behaviours within their user interfaces within stereoscopic rendering devices, such as VR and AR. We will be exploring an exponentially increasing level of degrees of freedom within how 3D buttons react to direct user interaction. Colour changing, moving, and deformable button reactions will be compared within a user study, where participants will interact with the varying modalities at different levels of overall effect. The buttons will be presented with low, medium, and high levels of their overall behaviour, encompassing different input data, while providing varying changes to their output effects. These buttons will be combined with mid-air contactless haptics to see what benefits may lie in conjunction with, or against, the varying levels, and modalities, of button reaction. We will give the designer a clearly summarised view of our results, as well as a full in-depth analysis and explanation of our results. Our findings indicate favourable opinions of simple reactions, with increasing complexity generally resulting in poorer opinions, ease of use, and overall enjoyment. Reactions that mimic that of the real world or 2D interfaces were mostly found to be more favourable than that of those fully utilising 3D principles such as deformation.

3.2 Background

Prior to our development and testing we examined a few key areas to gain a broader understanding of the problem at hand. Although the study was focused within the 3D realm, 2D interfaces were also taken into consideration. This was primarily due to the assumption that most participants will be heavily accustomed to digital 2D interfaces.

3.2.1 3D User Interfaces

Large amounts of research has been conducted into the implementation and usage of 3D user interfaces (LaViola et al. 2017) (Jackson et al. 2018) (Riecke et al. 2018). Rather than the conventional approach in 2D user interfaces that rely heavily on ray casting methodology (Wingrave and Bowman 2005) of clicks or touches to perform an action, the 3D realm allows for physically grounded approaches (Bowman et al. 2006). Treating the interface as part of the digital world instead of another 2D plane within the system increases the flexibility, but also the complexity of the interface.

It has been discussed that design principles that are leading towards a more natural and realistic approach tend to yield good results, while employing techniques that are only possible within the 3D digital realm may both enhance them, and also hinder them (Bowman et al. 2012). Other research suggested that interfaces could build upon the real world principles, enhancing them to create an improved reality (Stoakley et al. 1995a). We intend to make use of these naturalistic approaches and theories to interface design, while expanding on them to see what benefits may be presented.

3.2.2 Virtual Reality Interfaces

Virtual reality interfaces have the benefit of having depth perception within their medium, admittedly with caveats (Batmaz et al. 2019), allowing them to have greater flexibility when compared to many other mediums (Weiß et al. 2018). Designers have made use of this freedom successfully as done within the Daydream VR platform (McKenzie and Glazier 2017), while also being able to make use of both 2D and 3D medium.

Compared to conventional 2D displays, VR devices generally include displays with considerably faster refresh rates. Many standard 2D displays tend to have refresh rates around 60Hz, compared to VR displays that range anywhere from 72Hz (Oculus 2019b), to the more common 90Hz (HTC 2016a), and in a few cases up to 144Hz (Valve 2019a). These higher refresh rates not only benefit the user's experience with the simulation (LaViola 2000) (Claypool et al. 2006), proving a heightened level of immersion (Kim, Baddar, et al. 2017) (Ryan 2015), but have also been proven to increase the overall enjoyment of a user interface (Huhti 2019). Apart from the reduced latency, other benefits include heightened perspective motion understanding, as well as improved interface transitions. Underlying principles behind this have started to proliferate throughout other markets in recent years, with several new mobile platforms including devices increasing their maximum refresh rates from the market standard of 60Hz to 90Hz and even 120Hz in certain cases (Perry 2020).

However, virtual reality has its own challenges. The lack of ubiquitous physical input methods present interesting challenges (Kim and Choi 2019a). Many systems tend to rely on devices that the user hold in their hands, several being tracked with six degrees of freedom (6DOF). These devices often provide a good amount of user freedom, but tend to lack on input precision and flexibility (Holderied 2017). The resolutions of many current generation headsets are too low to support text at distance, making several interface choices challenging (Thompson 2016).

3.2.3 Hand Tracking

Controller free hand tracking allows the user to interact with the computing system, simply be using their hands. Many of these devices utilise standard RGB cameras (Panteleris et al. 2018), while others rely on using IR cameras and illumination (Erden and Çetin 2014). While the underlying technology tends to be different, the overall optical methodology and principles tends to be the same. Research has been conducted for well over two decades into the field, ranging from pose or gesture estimation (Sato et al. 2000), through to bone position estimations (Horowitz 2014).

Figure 11 The Leap Motion Controller, shown in the center of the image to the left of the keyboard, demonstrating hand tracking. (Ultraleap 2013)

Hand tracking hardware and software has been commercially available for a number of years. Devices such as the Leap Motion Controller (Leap Motion 2013), shown in Figure 11, and Microsoft Azure Kinect (Microsoft 2019a), and prior iterations, have allowed users to track their hands (Hald 2013). Several computing platforms are now starting to integrate hand tracking directly into the hardware of the device such as: the Oculus Quest (Oculus 2019b) Figure 12, Hololens 2, and Varjo VR-2 Pro (Varjo 2019), removing the necessity for users to obtain additional hardware. This seamless integration will not only help improve the adoption but also massively increase the user base of the tech.

This image has been redacted for copyright reasons.

Figure 12 The Oculus Quest VR headset has built-in experimental hand tracking, removing the need for external peripherals. (Oculus 2019b)

Optical hand tracking solutions have multiple current issues. Image based approaches still require some form of estimation (Pan et al. 2010), especially when it comes to occlusion of the hands and field of view (Bachmann et al. 2018). While users may not have to wear or hold anything, those options generally still offer full joint or position tracking, no matter how far away from the "origin" point of the tracking. Current implementations have been found to be worse than in certain task situations than standard VR controllers (Caggianese et al. 2019).

3.2.4 Mid-Air Contactless Haptics

Ultrasound mid-air haptics have started to become commercially accessible in the past few years, with off the shelf devices available to buy, admittedly though still within a niche market. These devices use focused ultrasound to produce a small region of pressure (Carter et al. 2013a). This focused pressure point, more commonly known as a focal point, is then moved in 3D space at high speed to create a path (Frier et al. 2018). Paths are then felt, specifically on the user's palmar surface where there is a greater sense of touch.

Figure 13 Visualisation of how a focal point is created using multiple ultrasound transducers in phase. (Ultrahaptics 2017)

Although these devices allow for intricate and highly detailed levels of control, they tend to lack in the amount of force applied to the user. Devices tend to be of relatively small size which affects the "interaction zone" in which they can be used. These notable problems come with the unique benefit that the haptic feedback can be applied without the use of any form of contact device.

Paired with an optical based hand tracking solution these allow the user to be fully hands-free from any devices. Benefits are only compounded when virtual reality is taken into account, where the user does not need to worry about holding or wearing a device for prolonged periods of time. Hygiene is a given, as these devices do not require the user to be touching or grasping anything to feel the effect of the device.

3.2.5 Hypothesis

We hypothesise, based upon the background research, that there is a distinct link between naturally reacting interfaces and their overall perceived performance (Kamel Boulos et al. 2011) (Falcao et al. 2015). These natural reactions replicate real world phenomena, which in turn should result in greater efficacy when paired with optical hand tracking. This would allow the user to directly interact with the interface, rather than through a proxy (controller). However, we also believe that several less explored options that exploit the usage of 3D depth may present interesting, and mostly unexplored results. Implementing mid-air haptic technology should allow us to explore how users respond beyond visual modality changes.

From this we have several questions and hypotheses that we intend to research.

- Q1. What is the effect on user opinion and preference when visual modalities of button reactions are modified?
- Q2. What is the underlying effect on user opinion and preference when haptic feedback is applied to the user when interacting with buttons?

Based on these questions we hypothesised that:

- H1. Colour reaction buttons would be the most favourable of the three options, due to their common place usage throughout 2D interfaces.

H2. Deformation buttons will provide too much feedback for the user, to the point that they are perceived negatively.

H3. The introduction of haptic feedback should provide an overall positive effect for the user.

Although there are other stronger haptic devices available, which can often provide haptics effectively beyond just the palmar surface, these generally present several issues when utilising optical hand tracking. Most devices require direct contact or attachment to the hand, such as those in a controller, which in turn increases the overall "friction" for use. This friction comes from usually requiring extra wiring to either power the device or provide data; or introducing extra occlusion of the bones of the hand, which in turn reduces the quality of the hand tracking. Several devices such as localised linear actuators are applied directly to the skin, which requires prior setup to ensure the correct position of the haptic device, that once applied cannot be moved. Limitations in physical movement either result in one of two scenarios, having a device that is too small to sufficiently cover the desired region, or having a device that is significantly sizable.

3.3 Methodology and Protocol

Our study was designed to test three key different types of 3D button visual reactions, these categories were: colour, movement, and deformation. These tests would help us understand which modality changes were most beneficial to implement within an interface. For this, a VR simulation was developed where users would perform simple interaction tasks across three distinct scenes, encompassing the three different types of visual reaction. Surveys were created for the pre, during, and post simulation stages. The pre-simulation survey would help us understand different user demographics, as well as helping us understand certain user choices. During simulation survey consisted of button ratings, and haptic ratings conducted through interactable sliders. The post-simulation survey queried users' opinions further than that of the simulation, allowing us to gain a deeper understanding of their emotional and technical responses to the simulation.

Within our study we will be implementing VR headsets, mid-air ultrasound-based haptics, and optical hand tracking. The combination of these three distinct technologies has had prior research conducted (Georgiou et al. 2018), however, few studies have directly focused on the user interface.

3.3.1 Study Development & Setup

We used the Unity game engine to create our VR simulation, and the HTC Vive headset as our output source. Rather than using traditional VR controllers, we made use of a Leap Motion Controller (SDK 4.4.0) which allows the user to interact with the simulation using their hands directly. By utilising this we can understand how the user would naturally, and physically, interact with the interface, rather than through a proxy device. The Leap Motion Controller was attached to the front of the headset, this was used for the visual representation of the hands, as well as the interaction logic. The Leap Motion interaction engine (version 1.2.0) was used for several core physical interactions, as well as a custom pose estimation system that would allow us to analyse the users hand when interacting.

To imbue a sense of physicality to the experience, we utilised the Ultrahaptics STRATOS Explore ultrasound haptic array which provides mid-air haptics to the user's hands. These devices create a focal point of high-pressure ultrasound at a point in 3D space, which is then moved at high speed to produce a path. Displacement of skin occurs on this path and the micro movements within the skin are felt (Frier et al. 2018). Haptic effects were applied when the user interacted with different buttons within each scene, tracked to their palm. There was a secondary Leap Motion Controller attached to the haptic array. This was in a fixed position which provided greater accuracy for positioning haptics.

Participants were sat down in front of a standard height table where the Ultrahaptics array was positioned, as shown in Figure 14. Each participant wore the HTC Vive headset throughout the entirety of the simulation.

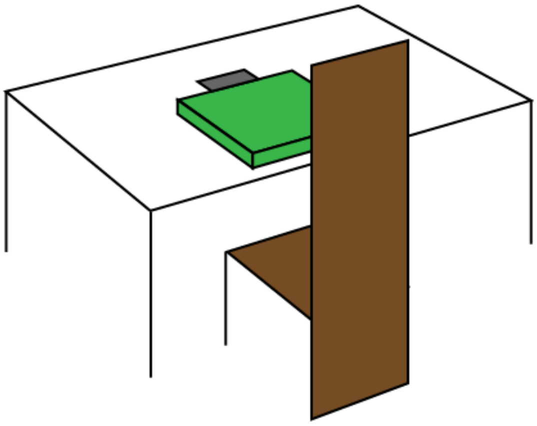


Figure 14 Diagram showing the physical setup of the simulation. The user positioned upon the chair (brown), in front of the haptic array on top of the table (green). One Leap Motion Controller (grey) was positioned at the top edge of the haptic array, with another on the front of the HTC Vive VR headset.

During the study, a number of different pieces of analytical data was recorded on each user. We recorded ratings, timings, and a video recording (including audio from a microphone) of the user's interactions. Video recordings were from a fixed digital viewpoint.

We followed the Ultraleap ethics principles throughout our study. All data from the study was completely anonymised. All users were informed that their video and audio would be recorded during the study.

3.3.2 Button Types

Each of the scenes within the simulation had three levels of reaction. Low, medium, and high-level reactions were applied to the buttons and can be broken down into their core underlying methodology. These three levels were designed in such a way where they would distinctly represent increasing amounts of stimuli for the user, and implementation complexity for the developer. This can be summarised as continuous increases in the degrees of freedom that a button could alter and represent.

Colour buttons operated with low being instantaneous colour changes between two, medium being a transition between two colours over time, and high transitioning between three colours with one highlighting before the user interacted (when in proximity).

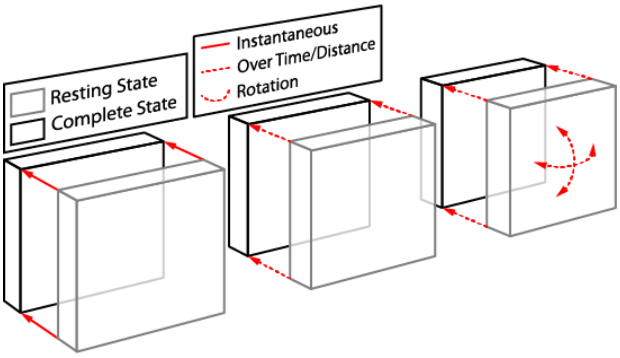


Figure 15 Diagram showing the three different levels of reaction for the movement buttons. Each button would be present within the scene.

Movement buttons would work in a similar fashion. Low reaction was an instantaneous translation along the Z axis. Medium would translate based upon the user's contact point. High level would rotate when in near proximity and then continue the same as the medium.

Deformation was the outlier as it had a considerably higher degree of freedom than the other set of buttons. Low reaction would instantly expand in size when interacted with. Medium reaction would magnetise to the finger in likeness to that of a Ferro fluid, a liquid that contains magnetic material that will attract and morph itself towards magnetic forces (Wikipedia 2021b). High level reaction would magnetise to the finger when the user is in close proximity of the button.

3.3.3 Button Interaction and Haptic Feedback

Low level reaction buttons would instantaneously react, with a binary reaction, changing from one single state to one other. Medium level reaction buttons would transition from one resting state to another interacted state, with a continuous reaction. High level reaction buttons would both pre-empt the user's interaction, as well as transition from their resting state to the interacted state.

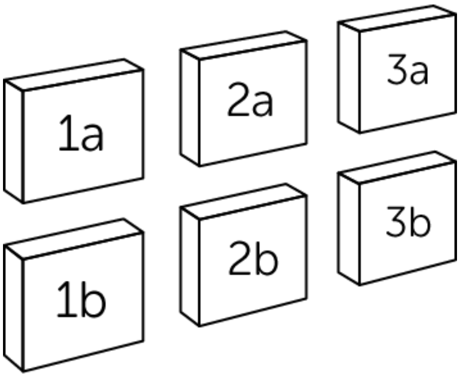


Figure 16 Diagram showing the button positioning as they would appear within the scene. The button order would be vertically randomised per scene (signified by 1-3), with haptic buttons being either the top or bottom of each vertical pairing (signified by a-b).

Haptics were also varied with level of reaction for half of the buttons within each scenario. All three haptic effects would produce the same shape and position, of a 2cm radius circle on the palm. This shape and position was chosen as the haptic effect is strongest at that point on the hand (Wilson et al. 2014). Although haptic positioning and contact points are notably different, they may still yield favourable results due to the visual stimuli taking precedent.

Low level buttons would play a 0.2 second full intensity haptic. Medium level buttons would ramp from the lowest perceivable intensity to the highest in tandem with their visual reaction. High level buttons would ramp their intensity as in the medium, however, would also then create a 0.2 second ramping "click" once the button interaction result was fully achieved.

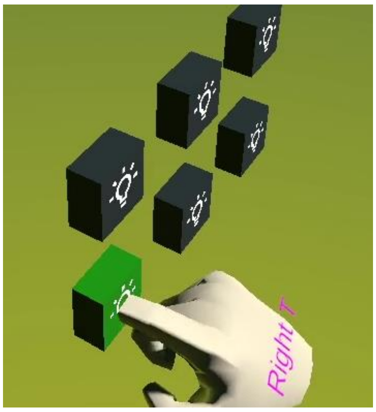


Figure 17 A participant interacting with the colour reactive buttons.

These three different types of modality change were chosen as they summarise three different types of digital representations. Firstly, colour is grounded solidly within the digital and electrical realms. The degrees of freedom are relatively simple, with many computing applications opting for colour shifts for their response design. Real world buttons tend to refrain from using colour on the button itself, often opting for a secondary reaction method of light or colour, disparate from the button.

Movement buttons ideals were heavily grounded within the real world, replicating physical response. Light switches, television remotes, and computer keyboards all use physically moving buttons with great success, applying that same knowledge to digital buttons should hopefully yield good results. The increased freedom of the digital realm allows us to increase the degrees of freedom available with these buttons, allowing us to try to more complex movements.

Deformation is a highly digitally grounded principle, with many real-world examples being found within the outer realms of the daily norm. With an exponentially higher degree of freedom than any other button before, it pushes beyond what is normally found within interfaces. Whether users are fond of these mechanics within interfaces is a desirable test we wished to explore.

3.3.4 Study Methodology

The simulation was developed as three separate scenes, with colour, movement, and deformation, each being their own individual scene. Users would interact with six different buttons within each scene, all of these buttons performing the same global action of turning a virtual light on or off. Half the buttons in the scene would have haptics that would be applied on contact. Once the user had interacted with the buttons, they would be asked to first rate their opinions of the buttons themselves, then secondly their opinion of the inclusion of the mid-air haptics for the buttons in that scene. This process would be repeated throughout all three different scenes and reaction types.

We opted for using in-simulation data collection as well as pre and post simulation surveys, with users rating their opinions of the buttons directly within the simulation. After the initial interaction stage, each user was provided with a set of ratings sliders, these ranging from 1 (best) to 7 (worst). Users had to grasp a block on the slider, which would then snap to the different rating positions based on proximity to them. By including these rating questions directly within the simulation we hoped to improve the overall immersion of the user, as well as collect more consistent data.

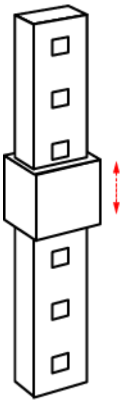


Figure 18 Diagram showing the rating sliders visual appearance. The red arrow denoting how the interactable portion of the slider would move, with smaller squares on the slider visualising the positions of the scale. The slider would have 1 at the top, which would be

signified in green, and 7 at the bottom signified in red. Values on the slider would be gradiented from green, to yellow, and finally to red.

To ensure a consistent set of data from each user we randomised the order of the scenes, as well as the positions of the buttons within each scene. Buttons were vertically randomised, then horizontally to ensure haptic buttons were in the same region (see Figure 16). This was done to ensure that the haptic effects produced by the array would be consistently felt across all three set of buttons. The Ultrahaptics STRATOS Explore has an "interaction zone" of around 20cm from the edge of the device's ultrasound transducers, thus limiting our possible haptic application area.

3.3.4.1 Study Timings

Our study was time limited across each three of the scenes. Users has a fixed two-minute time limit in which they were allowed to simply interact with all of the scene's buttons. If the user expressed their desire to continue prior to the two-minute time limit they were permitted to do so. After this segment, they were then given an unlimited amount of time to then rate the buttons. During this rating period they were still permitted to interact with the buttons, this was done to ensure the user would be able to correctly rate each button. Times were recorded for the interaction stage, initial button rating stage, and secondary haptic rating stage.

3.3.4.2 Study Surveys

We required participants to complete two different surveys in our study. The first survey was conducted prior to their experiences in the VR simulation, with the second occurring afterwards.

The pre-study survey was undertaken anywhere up to one week prior to the VR simulation, the user being free to complete the survey in their own time. This survey gauged the participant's demographic information such as: age, biological gender, VR usage, mid-air ultrasound haptic usage, and technology daily habits. These were recorded primarily so that the biological differences for haptics could be accounted for (Abdouni et al. 2017), as well as garnering a better insight into what interfaces the participant is familiar with. We queried the user's use of both virtual reality devices and mid-air ultrasound haptic devices to ensure we received a good distribution of users. Each user expressed their consent to the study, both during the pre-study survey and before they were about to perform the simulation.

The post-study survey was conducted directly after the user had completed the VR simulation stage. Users would be asked about four key areas: their enjoyment and ease of use of their interactions with the buttons, their interaction methods (hand vs fingers), their opinions on haptics, and their opinions and ideas on the study as a whole. The first two sections of the survey were primarily quantitative answers, while the later sections were entirely qualitative. Participants were not required to answer every qualitative question but were required to answer all quantitative.

To ensure users understood which buttons they were giving answers to, we included short repeating videos of the interactions of the different buttons. The buttons were not labelled low, medium, high, just simply listed as 1 to 3. Ratings were recorded on a 7 stage Likert scale, while opinion questions were recorded as standard text.

The full set of questions can be found in the appendix, for both the pre study survey (7.1.1.1) and post study survey (7.1.1.2).

3.3.4.3 Participants

We recruited 23 participants, 19 male and 4 female, from within the Ultraleap office staff. Each participant was recruited with no prior knowledge of what the study entailed. All users signed consent to be taking part in the study, as well as having their simulation video and audio recorded. The study was approved by the Ultraleap ethics committee, following their consent structure.

Our participants had an average age of 30.56 years, ranging from 22 to 47. All bar one participant had used VR and mid-air haptics at least once. 26% of the participants were regularly using (at least once a month) VR, while 70% were using mid-air haptics. There was an almost equal split between the usage of Windows or Mac within the participants, 43% to 57% respectively. 91% of participants said that they used smartphones as one of their primary technology devices outside of work, this was followed by laptops at 61%. Users could express as many choices as possible here, from a list of smartphone, laptop, desktop computer, tablet, TV, and car. 65% of participant's jobs were within the field of software research and development. These surveys were used to gain an understanding in the prior experience with different interfaces.

3.3.4.4 Data Analysis

Data analysis was performed within the Python (Python 2021) scripting language, utilising the jupyter (Jupyter 2021) environment. This was used to both analyse our results, as well as produce graphs and visualisations.

We performed our data analysis with two sets of data, first with the full set of all participants, and secondly with a smaller subset of data where a few sets of users were removed. This was done to ensure validity of our results and to ensure our dataset was more robust. We chose to remove two users who reported highly repetitive scoring across the study, both during the simulation and post study as these were reported as outliers. We removed two sets of users, primarily due to haptic reasoning. Firstly, female participants were removed due to differences in tactile perception (Abdouni et al. 2017), however, this was also done due to too small a sample size to effectively understand any present gender effects. The second group being any participant over 40 years of age as this has been shown to be an age of tactile transition (Abdouni et al. 2017) (Reuter et al. 2012), in this case that happened to be one participant. This resulted in a subset of 16 users. Analysis of this subset was not found to be significantly different to the full set, thus we decided to retain the usage of the full data set for our analysis.

3.4 Results

The results are explained in the following sections, where we will cover the participant feedback and their interactions within the study. Core button statistical data will be divided based on reaction modality, split by colour, movement, and deformation. For each of the modalities, the three levels of button were independently queried. This forms a three-by-three analysis of modality by reaction level. Each participant was queried in their opinion of the different buttons during the study on a Likert rating of 1 through 7, with two follow up questions in the post-study survey querying their perceived ease of use and enjoyment of the buttons. Haptic effects were analysed against their modality, creating a single three-way analysis. The haptic opinion was recorded during the study, directly after the button opinions.

A large majority of our data was shown to be non-normally distributed between our varying factors and effects. To account for this, we applied an Aligned Rank Transform (ART) to our data, for both the button and haptic opinions. These transformed values were then analysed using factorial non-parametric repeated measures analysis of variance

(ANOVA) tests. By performing these transformations and tests, we could analyse the interactions between our factors, as well as the overall main effects.

Statistics will be reported for each of the independent variables and factors. These will include their geometric means (gM) and geometric standard deviation (gSD). Geometric values are reported due to the data being transformed by the ART. Significance values will be reported, along with any respective interactions found within follow-up pairwise t-tests. All dependent variables will have their values reported, coupled with their data skews. Dependent variables that are not transformed through ART will have their normal means and standard deviations reported.

Graphs will be displayed for the overall independent variables, with significant interactions displayed highlighted. These graphs will be displaying ordinal data, and as such will be visualising the geometric means and geometric standard deviations.

3.4.1 Scenario Quantitative Results

3.4.1.1 Button Ratings

Overall rating values were $(gM = 2.806, gSD = 1.834)$ with a slight positive skew of (0.351).

Main effects of button reaction modality and reaction level were compared to understand their effect on the participants overall opinion of the button. No significant 2-way interaction was reported. A significant interaction was reported for the modality $F(2, 44) = 3.734$ $p = 0.032$, however, follow-up pairwise analysis did not report any significance. No further significant interactions were reported.

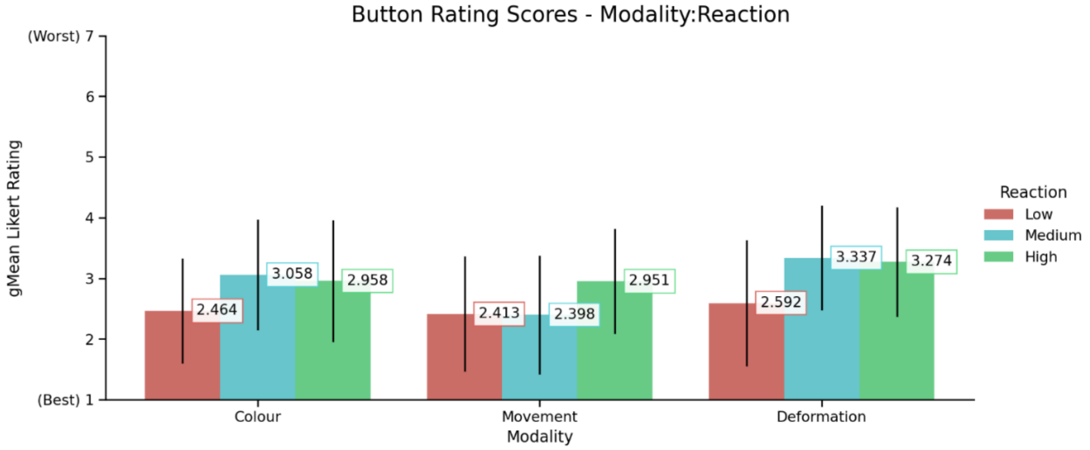


Figure 19 Graph showing the geometric mean and geometric standard deviations of button rating scores, grouped by button modality and reaction level.

Modality	**Colour**			**Movement**			**Deformation**		
Reaction	Low	Med	High	Low	Med	High	Low	Med	High
gMean	2.464	3.058	2.958	2.413	2.398	2.951	2.592	3.337	3.273
gSD	1.697	1.785	1.975	1.869	1.923	1.704	2.045	1.698	1.777

Table 1 Table showing the geometric mean and geometric standard deviation values of button rating scores, grouped by button modality and reaction level.

3.4.1.2 Button Ease of Use

Overall ease of use values for the buttons were $(gM = 2.268, gSD = 1.742)$ with a positive skew of (0.712).

Main effects of button reaction modality and reaction level were compared to understand the effect they presented on the ease of use of each button. Significance was found with the 2-way interaction of modality and reaction $F(4, 88) = 2.828$ $p = 0.029$, however, no follow-up pairwise t-test reported any significant interactions. The reaction main effect showed significance $F(2, 44) = 11.218\ p = < 0.001$, with corresponding pairwise t-tests reporting a significant interaction where the low reaction buttons were easier to use $(gM = 2.079, gSD = 1.703)$ than the high reaction levels $(gM = 2.467, gSD = 1.832)$. No further pairwise t-tests showed any interactions. The modality main effect did not show significance.

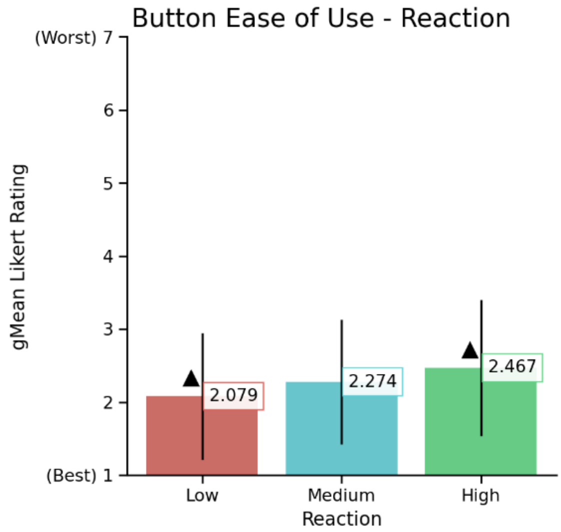


Figure 20 Graph showing the geometric mean and geometric standard deviations of button ease of use, grouped by reaction level. ▲ denotes significant pairwise interaction.

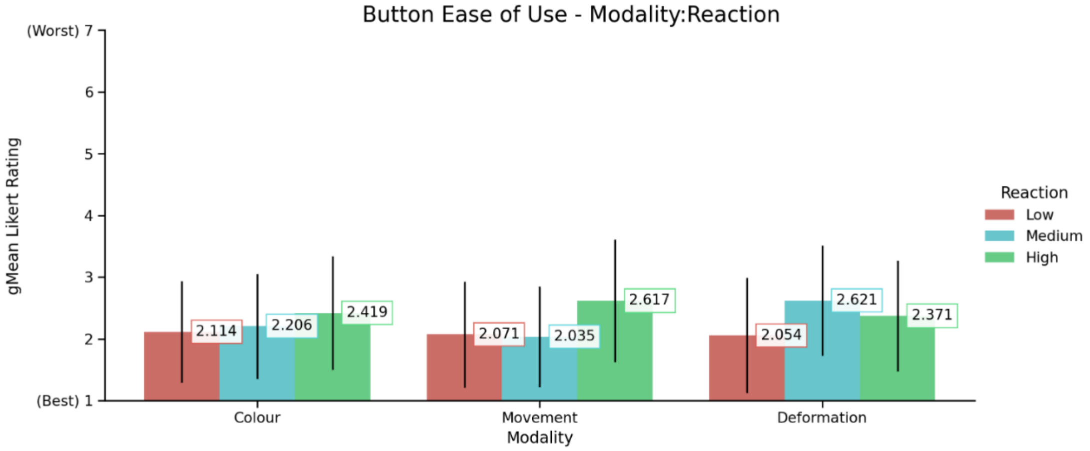


Figure 21 Graph showing the geometric mean and geometric standard deviations of button ease of use, grouped by button modality and reaction level.

Modality	**Colour**			**Movement**			**Deformation**		
Reaction	Low	Med	High	Low	Med	High	Low	Med	High
gMean	2.114	2.206	2.419	2.071	2.035	2.617	2.054	2.621	2.371
gSD	1.620	1.668	1.808	1.686	1.603	1.961	1.836	1.752	1.764

Table 2 Table showing the geometric means and geometric standard deviations of the button ease of use, grouped by button modality and reaction level.

3.4.1.3 Button Enjoyment

Overall user enjoyment of the buttons was reported as $(gM = 2.379, gSD = 1.668)$ with a slight positive skew of (0.276).

As with the other values, analysis was conducted between the main effects of button reaction modality and reaction level to understand their effect on the overall enjoyment of the buttons. No statistically significant interactions were reported for either the 2-way interaction or either of the main effects.

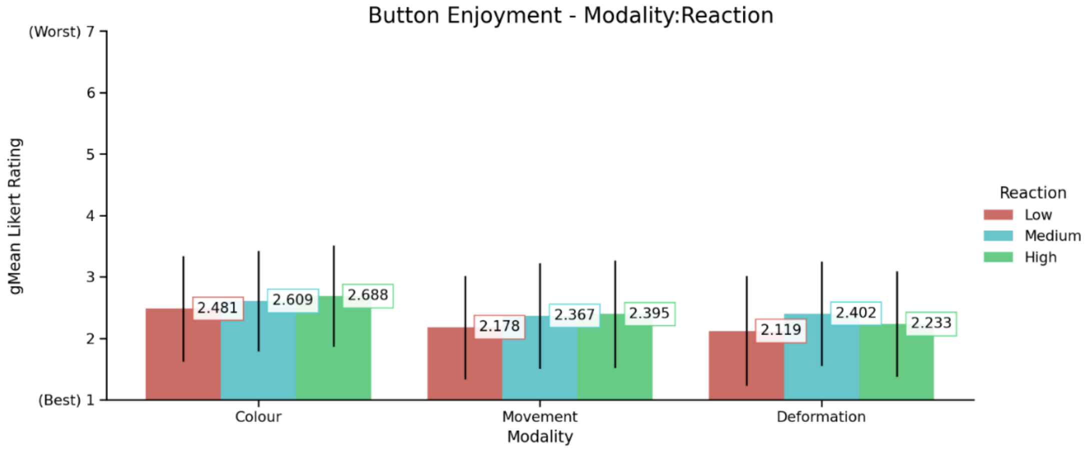


Figure 22 Graph showing the geometric mean and geometric standard deviations of button enjoyment, grouped by reaction level.

Modality	**Colour**			**Movement**			**Deformation**		
Reaction	Low	Med	High	Low	Med	High	Low	Med	High
gMean	2.481	2.609	2.688	2.178	2.367	2.395	2.119	2.402	2.233
gSD	1.680	1.606	1.613	1.649	1.685	1.720	1.757	1.674	1.687

Table 3 Table showing the geometric means and geometric standard deviations of the button enjoyment, grouped by button modality and reaction level.

3.4.1.4 Haptic Ratings

Overall haptic rating of the buttons was reported as $(gM = 2.718, gSD = 1.827)$ with a positive skew of (0.590).

We compared the main effect of button reaction modality to understand the impact haptics had on the overall opinion and perception of the different button types. No statistically significant interactions were observed for the haptic ratings.

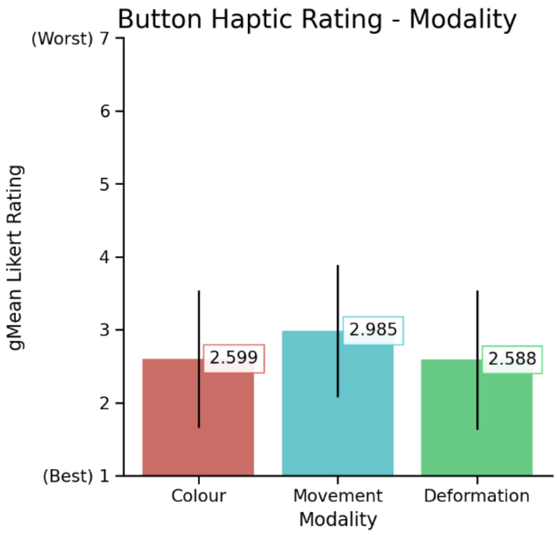


Figure 23 Graph showing the geometric mean and geometric standard deviations of the effect of haptics on buttons, grouped by button modality.

Modality	**Colour**	**Movement**	**Deformation**
gMean	2.599	2.985	2.588
gSD	1.846	1.78	1.882

Table 4 Table showing the geometric means and geometric standard deviations of the effect of haptics on the buttons, grouped by button modality.

3.4.2 Interaction Methods

Through the video recordings, poses and interaction methods were recorded and analysed. We recorded two different metrics: the first point of interaction for that scene, and the most common method of interaction for the scene. These interactions were classified in three distinct ways. Index finger touching was when the user was touching the object with just their index finger, multi-finger touching was when the user was using multiple fingers (2 or more) to interact with the button, and finally hand/palm touching where the user was attempting to use their whole hand to touch the button.

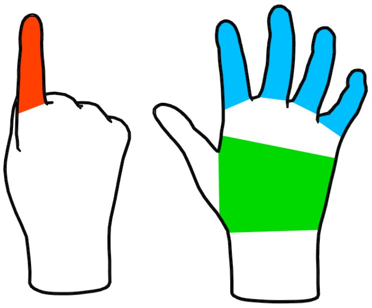


Figure 24 Diagram showing the differences in hand interaction points. The hand on the left shows the index finger "pointing" region in orange. The hand on the right shows the open palm options where blue signifies multiple fingers touching, and green signifying whole hand touching.

Participants showed a clear majority towards interacting with their fingers, with an average of 81% of users opting to use their fingers for their very first interaction of the study. This trend continued throughout as it rose to 84% for first interaction across all scenes. Users were consistent in their interaction methods, with 92% of them opting for using their fingertips. Breaking this down further, 44% percent of these were using just the index finger, while 48% were using multiple fingers.

Survey recordings asked users to respond with whether they preferred: finger touching buttons, or whole hand touching buttons. The trends observed in the study were mirrored closely in the post-study survey, with 71% of users expressing that using one or more of their fingers to interact with the buttons was the easiest method to do so. 29% of users felt that using their whole hand was easiest.

3.4.3 Completion Times

Users completed the entire study with a mean time of 6 minutes 57.633 seconds, with a standard deviation of 1 minute 54.953 seconds. These times were for the whole study, incorporating the button interactions and ratings portions of each trial. The table below shows the means for the individual trials and their standard deviations.

Modality	**Colour**		**Movement**		**Deformation**	
Value	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
Interaction Time	83.231	27.775	86.123	32.153	75.545	33.954
Rating Time	43.727	32.085	46.148	49.740	30.477	26.296
Haptic Rating Time	18.176	17.094	20.873	30.350	13.322	9.175

Figure 25 Table showing the differences in geometric mean and geometric standard deviation times for the different stages of the study in seconds.

3.5 Discussion

As stated through this paper, we were exploring the levels of effort required to provide and enhance effective 3D user interface reactions. The overall goal being to deliver a set of guidelines that help the interface designer to make informed choices as to how to improve their interface or, counter-actively, reduce the required amount of development to achieve beneficial results.

While the study attempts to provide a significant insight into user interface choices and design, it has a number of limitations. Firstly, due to the nature of user interfaces, we can only really provide insights into a small subsection of interfaces and their design. It would not be factual to present these findings as fully concrete evidence for their usage throughout user interface designs as a whole. Secondly, the technology used required several limiting factors, such as overall movement range and compatibility between desktop (the mid-air haptic array) and VR setups. Coinciding with this, the results for haptic feedback may not remain completely relevant in a few years' time. This is due to the quickly evolving and changing technology that drives the haptic feedback. Third and finally, the interface reactions that are covered here may no longer be entirely problematic as once thought in a few years time. It is entirely plausible that with increases in processing and computing power, the inclusion of real time deformation or similar shape changing effects creates a scenario where people are far more accustomed to it.

We quickly found trends when specifically looking at just the scenario quantitative results, however these trends were also backed up by the supporting time, background, and qualitative data. The lowest reaction colour button was shown to be the unanimously highest scored item across all user groups (both with outliers and specified users removed and without), it had the smallest standard deviation that helped cement this. Users commented throughout the post-study survey that they felt this button being the most indicative of how the button should be reacting. Not only was it the best rated, but it was also similarly one of the easiest buttons to use. Both score and ease of use had relatively similar, linear, negative trends as the complexity increased, with both their mean and standard deviation increasing. This could be attributed to either the speed of transition, or the method of visual change being disparate to the interaction effect. Participants commented that although the medium and high-level buttons were more engaging and interesting, they were less representative of their expectations. Even though the lowest reaction presented the best overall version of the group, it was often reported as being too basic and the least fun to use.

Unlike colour, movement showed uniform good scores and enjoyment for all buttons, although score had higher standard deviation. These buttons effects were grounded more towards the real world in their effect, which most likely reflected well in their ease of use. Both the low and medium reaction buttons worked similarly by translating along one axis, however the added rotation of that axis severely affected ease of use. Many participants commented that the high level was too complex or challenging to use, with one user commenting they disliked the feeling of “being watched”. Participants agreed that these buttons were of a highly natural expression. These results were mostly expected, as our background research had expressed natural and real-world based interactions would generally yield favourable results (Malizia and Bellucci 2012).

Deformation was entirely different to the prior, with users enjoying and understanding the lowest reaction button, but struggling to comprehend the medium and high reaction. The degrees of freedom for the buttons were a magnitude higher than any of the other buttons in the test, with any individual region of the button being able to move, something that is not generally found within the real world to a large degree. We expected this to a degree, with prior research suggesting that effects beyond realism can be shown as confusing to the user (Bowman et al. 2012). Participants had widely varying responses in the survey, with several finding the buttons incredibly fun to interact with, while others not understanding what exactly the buttons did. There was a consensus that the buttons were too complex for the current task at hand, presenting too much visual information compared to their result. However, this was also conflicted with several users expressing that the button reactions were too subtle.

The addition of haptics to several of the buttons presented favourable responses from participants. Results showed trends towards heightened engagement and a general positive feeling for their inclusion. Comparatively users did report that the positioning was not linked to their visual expectation, something we noted may be a problem prior to the study. Nearly all participants could tell the differences between haptics, with many commenting they were wishing for greater force. While this is not entirely surprising, the frequency of these comments could well have been considerably higher if we had not chosen to use the palm positioning.

Many of the buttons increased degrees of freedom lead to a common trend in user opinions. As the enjoyment and fun the user had from the button would generally decrease as the level of reaction increased, their ease of use would generally follow suit. Having buttons that are fun to use on their own is fine, however when implemented into a scenario where the buttons are not the key and thus ease of use should take priority.

Timings of the scenarios helped solidify the choices that were being made by the users. Not only were users unanimous in their choices for the deformation, their timings for doing so were also considerably lower than the other scenes. In this vein, it was easier for users to provide a less favourable opinion of buttons, however, took greater consideration of when they were of merit. This was reflected within the movement scores, where they were considerably better, with rating times having incredibly high standard deviation.

Interaction methods yielded interesting results with users, especially when taking into account the haptic interactions. Generally, current mid-air haptics require relatively large amounts of the palm present to be truly effective (Wilson et al. 2014). This means index finger pointing (where the hand is mostly curled in on itself), will deliver poorer haptic results as this often covers the majority of the palm. Movement scenarios had an overall poorer average score for haptics, and in turn had a higher percentage of users opting to use their index finger for interaction. Comparatively, colour had a better average haptics score, and users were found to be more likely to use either their whole hand or multiple fingers (with their palm open). This could be due to the movement buttons objectively requiring a more precise interaction, due to their reliance on physically moving beyond the continuous plane of the other buttons.

User scores could also be attributed to the backgrounds of the participants, this being intrinsically tied to their differences in technology usage. As we found, the majority of users relied heavily on smart phones for their day-to-day technology needs, with only 26% of users using VR within the past month. This lack of VR usage could help attribute as to why they found colour buttons (that did not have any movement) to be the easiest to use. Many of the desktop and mobile interfaces tend to rely on quick, colour-based transitions with small amounts of movement.

3.5.1 Suggested Guidelines

Based upon the findings of our user studies and surveys, we can summarise the information into several key points.

First, that instantaneous reactions are generally going to be the easiest and simplest for the user to understand. These reactions should be strongly linked in temporal response, with the cause and effect happening together. Many users are used to this form of interaction and mantra within their daily technological usage, expecting the interface to respond quickly and in a timely fashion. This was validated with the opinions and ratings through the testing where users were almost unanimously rating them better. Haptics helped confirm this, with users preferring instantaneous feedback haptically as well as visually. These reactions will not only be some of the best, but also easiest to implement.

Second, that methods of reaction based upon physical, real-world ideas and principles generally elicit favourable opinions, both for ease of use and general user enjoyment. User opinions were favourable when they could place the reactions with something familiar or expected. Reactions that are beyond established norms can come across confusing unless their degrees of freedom are reasonably limited. Highly digitally grounded reactions are unfamiliar, however could be understood and enforced over time. Ensuring haptics are linked to your visuals will improve their effectiveness.

Third, that highly expressive interfaces, with high degrees of freedom, can detract from the experience at hand. Adding complex mechanics to your user interfaces not only increases the amount of required development, but then in turn detracts from the overall output. Although you want the user interface to be interesting and engaging, adding too much complexity will harm the user's enjoyment. They'll spend increased amounts of time simply trying to interact with the interface, rather than the experience itself.

Fourth, make use of depth where possible, users will instinctively and quickly understand what is happening. Translations and expansions upon the horizontal and vertical axes are effective at showing state changes, but less obvious in response to the user.

Fifth, ensure that your reaction state changes and resulting effects are intrinsically linked. All completion reactions should occur at the same as your action effect, with minimal delay. Having reactions complete either before or after your action will be confusing for the user.

Combining techniques outlined here is encouraged, however be sure to take into account the outlined points. Logically combining techniques should, by design, yield good results, such as having a moving button instantly change colour once fully reacted. Reversing that, having a button change colour over time and then move on completion should yield a poor result.

3.6 Conclusion

In this chapter we have studied and tested three different modality changes for 3D user interface button reactions within virtual reality when using naturalistic technologies. These modalities were guided by current technology and real-world mechanics for buttons, ensuring a highly comparable set of buttons when compared to widely implemented interfaces. We developed a testing environment that allowed participants to interact with and give opinions on each button. Our development encompassed several novel technologies, which have limited prior research when combined and within this context. We then analysed our testing results to give a deeper understanding and deduce trends. Finally, we present a set of guidelines through which we disseminate our findings to aid the interface designer for when creating 3D VR interfaces.

4 Comparing Input Methods and the Role of Haptic Feedback During VR Hand-Object Interaction Tasks

4.1 Introduction

Virtual reality (VR) and augmented reality (AR) devices are becoming considerably more ubiquitous, with both consumer and industrial devices reaching new levels of quality and efficacy. These improvements have been powered in-part by advances in processing power available, resulting in smaller, self-contained packages. Where once these devices necessitated external computers, simply acting as an output screen, newer devices are able to perform all their required calculations on their own. With these improvements come significant reductions in barriers to entry and friction in usage, but also a shift towards naturalistic input methods. Devices that formerly relied on specialised controllers are starting to relinquish their need, opting towards novel sensors.

Transitioning from these input methods summarises an industry paradigm shift, where the prior reliance on specialised tools is being directed towards to the bare human hand. Hand tracking has been available for a number of years in varying forms, and through varying sensors. While it may be able to replicate hands visually, providing robust gesture and pose recognition across multiple studies (Sagayam and Hemanth 2017) and frameworks (Zhang et al. 2020), pain points have historically been found within direct 3D manipulation (Li et al. 2019). As devices start to edge towards these more intuitive and realistic interaction methods, there are numerous questions and problems still to be answered.

Relinquishing the reliance on physically held or grasped devices improves the ease of access to simulations, however, it also significantly reduces the possibilities of providing haptic feedback from the system. As the user is no longer directly touching a device new methods are required to undertake this task. Different devices provide varying levels of success (Arafsha et al. 2015) and feedback types (Hoppe et al. 2018), but all try to maintain the core goal of not physically covering or impeding the natural hand pose. Such devices as mid-air ultrasound haptic arrays (Carter et al. 2013b) allow for the user to feel haptic feedback in 3D space, without having to touch anything. Combining these devices with optic hand tracking could provide beneficial improvements to interactions, replicating several of the removed feedback states.

Comparing technologies of these natures has occurred on several occasions (Masurovsky et al. 2020), however, these tend to be lacking in several key areas. Firstly, the tasks focus solely on the comparison of controllers versus hand tracking, without any form of haptic application being used. This is usually due to the lack of haptic feedback being available for hand tracking, thus not being able to provide comparable conditions. While secondly, the tasks that are used are not often replications of existing fine motor skill tasks, but simple object manipulation tasks. These tasks may suffice certain specific movements, yet they do not provide realistic interactions that would be present within many commercial simulations or real-world activities.

Within this chapter we will be covering the development and testing of digital replications of fine motor skill tasks within virtual reality. These tasks are designed to capture and test fundamental manipulations of objects, used throughout common interactions in a person's daily life. Our study directly compares the effects of two different input methods, the first being optical hand tracking, and the second through physical controllers. Haptic effects will be compared, with mid-air ultrasound based haptics being applied to hand tracking, and contact based linear actuator haptics for the controllers. Each of these factors will be both statistically and subjectively contrasted and analysed, to understand overall performance and general opinions of the varying input and haptic modalities. Our results show that hand tracking excels at grasping and reduces failure rates, while controllers produce improved release events. Haptic effects were generally less favourable than expected, with multiple factors performing worse than without, although still performed well under certain scenarios.

4.2 Background

4.2.1 VR Inputs

While currently available consumer VR hardware ecosystems often come equipped with specifically designed controllers for said product, there is an industry shift towards devices that utilize non-conventional controllers or introduce novel additions to their devices. These devices are attempting to reduce friction for the user by either being cheaper, wireless, or seamlessly integrated with the ecosystem, unlike that of older devices such as the Phantom Omni (Sensable Technologies 1994) or Razer Hydra (Wikipedia 2011). Many recent consumer controllers opt towards having either 3 degrees of freedom, tracking rotational movement such as the Oculus Go controller (Wikipedia 2021c), or 6 degrees, allowing for both rotational and translational movement such as the HTC Vive controllers (HTC 2016a), for the hand; however, the finer differences are when it comes to tracking individual fingers and bones. Rudimentary estimation of fingers and hand poses has been available within different consumer systems such as the Oculus Rift Touch (Oculus 2016) and HTC Vive controllers (HTC 2016a), with these systems replicating hand movements through capacitive touch events and buttons. More recent offerings replicate the users finger curl positions through IR and pressure sensors such as the Valve Index controllers (Valve 2019b). Specialized technologies such as motion sensing or "exoskeleton" based gloves, such as HaptX (Varga 2021) and Dexmo (Dexta Robotics 2021), allow for greater definition of the hand and finger translation, but often have to be positionally tracked via a secondary device such as a HTC Vive Tracker (HTC 2017). These devices generally require significant initial setup for the user and system, usually being hardwired to the system due to power or data requirements. Commercial camera-based approaches have provided hand tracking without having to physically hold a device, such as the Microsoft Kinect (Microsoft 2010), Leap Motion (Leap Motion 2012), and Oculus Quest (Oculus 2019a), but struggle with occlusion, overall field of view, and at times latency (Guzsvinecz et al. 2019) (Silva et al. 2013).

Comparing physical controllers to one another has happened on numerous occasions across the years as different devices have been developed (Llorach et al. 2014) (Boletsis and Cedergren 2019) (Moro et al. 2017) (Mayor et al. 2019) (Young et al. 2014) (Kim and Choi 2019b) (Coburn et al. 2017). Multiple comparisons have been made between different hand tracking technologies, such as the Microsoft Kinect (Aditya et al. 2018), Leap Motion (Guzsvinecz et al. 2019), Optotrak (Tung et al. 2015), and even marker-based mo-cap systems (Ganguly et al. 2021). Optical hand tracking, specifically Leap Motion devices, compared to more conventional controllers is a less explored area.

As the XR industry attempts to implement hand tracking in different sectors, there will be numerous questions about possible benefits, differences, and challenges surrounding it. Devices such as the Microsoft Hololens (Microsoft 2016) and Oculus Quest (Oculus 2019a) have already implemented hand tracking into their hardware, allowing the user to use the device without controllers. Both devices have had varying success and capabilities in their implementations, with the Hololens being regarded as relatively slow and simplified, and the Quest's focus on direct manipulation and tracking. As with the introduction of touch screens to mobile phones, many of the questions and best practices will not be found instantaneously, however, we can attempt to cover several of the core fundamentals and principles. These should help provide insights into the underlying interactions, without requiring the context of a specific product or application.

4.2.2 Object Interaction in VR

Direct object manipulation is the most logical approach when it comes to interacting with objects within VR. Unlike techniques such as Go-Go (Poupyrev et al. 1996), world in miniature (Stoakley et al. 1995b), it requires no fundamental prior understanding as the approach is the same in the virtual world as the real. The technicalities and challenges come from exactly how that process of grasping is achieved, with many systems relying on a single button and a relative position to grasp an object. While that is applicable when utilizing a controller or a binary "button" interface, the process of achieving similar results is significantly more challenging when working with a nuanced input, such as the hand. Development frameworks, such as the Leap Motion interaction engine (Ultraleap 2021a) and Microsoft mixed reality toolkit (MRTK) (Microsoft 2019b), have improved the ease of implementation and interaction when utilizing directly tracked hands, but questions still remain about their effectiveness compared to grasped controllers.

Controller based systems are often attempting to visually replicate what hands should be doing, usually by creating hand structures around the controller, such as the SteamVR skeletal input system (Valve 2018). These implementations tend to focus towards matching the hand pose to the object, even if the physical position of the hand is not physically possible. There is a clear desire and necessity for better hand representation within VR, on both a technical and visual level.

Several studies surrounding the usage of low-cost optical hand trackers have focused on gestures (Gunawardane and Medagedara 2017) or pointing (Bessa Seixas et al. 2015) for indirect manipulation, however, these studies tend to avoid direct 3D manipulation. Many of these studies are attempting to supplement or replace keyboards and mice based interactions. These studies result in fairly unrealistic comparisons, especially when nearly every VR device is shipped with some form of custom controller.

Studies have found differences between controllers and hand tracking devices, for items such as grasp time and release times of objects (Masurovsky et al. 2020).

4.2.3 Haptic Feedback

Contact based haptics have been implemented for many years with different technologies such as eccentric rotating mass vibration motor, linear resonant actuators, and piezoelectric haptic actuators. These devices often provide haptic feedback by being held, grasped, or attached to the user, with implementations being relatively common throughout many commercial controllers such as various VR controllers (Valve 2019b) (HTC 2016b), and games console controllers (Sony 1997) (Nintendo 1997) (Microsoft 2001). While these are perfect for physically held devices, they introduce a number of problems when working with optical hand tracking solutions. Contact-based solutions often occlude parts of the hand as they necessitate contact or attachment to the hand, and generally go against the ethos of optical tracking solutions where the user does not have to physically hold a controller. Optical hand tracking solutions can often struggle with contact based haptic approaches, as they generally obscure critical parts of the hand.

Ultrasound based haptics provide haptic feedback onto the user, without directly touching them. By focusing high amounts of pressure into a singular point in space, known as a focal point, contact-less haptic feedback can be created and felt on the palmar surface of the hand through displacement of skin (Carter et al. 2013b). By combining this technology with optical hand tracking solutions, the user can receive haptic feedback without having to wear or hold any device in their hand. The comparison of this technology with contact-based haptics has had limited research exploration. Prior implementations of this technology within virtual reality have been limited to only allowing hand tracking to be used within a small volume surrounding the device (Martinez et al. 2018).

Previous work into the rendering of 3D shapes showed benefits of rendering haptics during contact (Martinez et al. 2019), however, the objects were not interactable.

4.2.4 Fine Motor Skill Tasks

Fine motor skill tasks are simple tasks that require the usage of the smaller set of muscles found in the hands, to, generally, perform tasks of high precision (Wikipedia 2021d). They form one of two types of tasks designed to help analyse and measure the movements and general performance of different bones of the body. The contrasting set of tasks are that of gross motor skills, that focus towards larger body parts such as arms and legs. Both types of task are commonly used to help develop and understand the level of physical and mental development within a child, with tasks often being targeted for different age brackets. Fine motor skill tasks are often limited in scope, requiring relatively small amounts of overall travel for the participant, but involve some level of consistent focus to be efficiently completed. We chose to replicate these tasks as we were looking to understand core differences between our chosen input modalities, at an incredibly fundamental level. Many of these tasks allow for highly focused movement analysis, especially when the task is of low cognitive load.

Many tasks have been studied over the years, with multiple instances of skill learning (Levac et al. 2019) or patient rehabilitation (Cabrera Hidalgo et al. 2018).

While there are a large multitude of possible fine motor skill tasks available (Kid Sense 2011), the choice of which tasks to focus on came down to a few key requirements and technological challenges. We wanted to ensure that we were able to faithfully replicate the task within the simulation, in regard to logic, physics, input interaction, and reduced possible completion complexity. Coupled with this, we wanted to make sure the tests were as little about the participant having to solve a mental task, and purely about the process of movement and object adjustment.

Tasks where the participants have to draw or trace letters, shapes, or paths, were considered. Unfortunately, due to the nature of these tasks generally requiring the usage of a secondary tool, they would not deliver the insights into direct object manipulation we were looking for. Replicating a virtual tool properly with hand tracking is a different challenge, and while interesting would be beyond the scope we were looking to study.

Several possible tasks are simply modifications of prior tasks with either difficulty increases, or minor adjustments to the outcome of the task (e.g. stacking six blocks instead of three). Although these tasks may push the overall complexity requirements of the task, they detrimentally affect overall length of the task. Ensuring the task length was adequate would help us retain focus and attention of the participant, while still obtaining useful statistics.

As we intended to develop a real-time virtual reality study, there were certain limitations with the technologies that restricted our overall choice of tasks. Currently available games engines provide flexible working environments for developing VR simulations quickly, however, they generally trade full realistic physics accuracy for speed and efficiency. Physics calculations in most engines, and to that extension most generic input devices, do not handle small objects particularly well, with numerous instances of objects passing through others, especially in cases where dynamic physics adjustments are made, such as translation changes. This limited the overall size of objects we could test for, thus removing possible tasks such as threading beads onto strings.

4.3 Methodology and Protocol

Our study was designed to test how users interact with 3D objects when provided with simple cognitive or precision tasks. We wanted to create tasks that would help test our hypotheses, while still producing fair and comparable differences between test cases. To this degree we chose to implement two digital recreations of simple fine motor skills tasks. The first task was a block stacking task, where participants would stack blocks in descending size order within a pre-determined area, followed by dismantling the created block tower in the reversed order to stacking into a secondary area. The second task was an object sorting task based upon varying visual features of repeated identical objects. In this case, blocks of the same structural form, with either one, two, or three markers total on their sides, were sorted into specific containers. By using these tasks, it allowed us to explore commonly used, real-world foundational tests as a basis to benchmark a number of different input modalities and changes in haptic presence.

We developed a VR simulation where participants would undertake two distinct fine motor skill tasks, with four different input combinations of hand tracking, VR controllers, and the application of haptics to both, visually explained in Table 5, resulting in a 2x2 within-subjects design. A total of eight possible combinations were repeated twice, making a combined total of sixteen tasks per participant, with a randomised order for each participant. These tasks looked at two key areas, precision of interaction, and continuous pose movement. For each of these tasks, the participant would receive a randomized input combination, between either optical hand tracking or capacitive hand tracking achieved through VR controllers, and a haptic combination of with or without.

Task	Input	Haptics
Block Stacking	VR Controllers	Linear actuator contact haptics
		None
	Hand Tracking	Mid-air ultrasound haptics
		None
Object Sorting	VR Controllers	Linear actuator contact haptics
		None
	Hand Tracking	Mid-air ultrasound haptics
		None

Table 5 The outlined matrix of possible task combinations. Each of these were performed by the participant twice.

Each segment of the study was recorded through a custom in-simulation recording system, called PlayRecorder which is described in full in the PlayRecorder chapter. This allowed us to store a one to one digital representation of each participant's playthrough for later playback, including all positional and interaction information with every object within the task. These recordings would be used to create quantitative data that we would then be using for our analysis. Recordings were created when the participant started a new test condition, and then saved on condition completion. This resulted in sixteen recordings per participant.

Prior to undertaking any task, each participant was required to perform a simple acclimatisation stage. This segment would teach the participant about how to successfully grasp and hold objects with the varying input methods. Two objects were presented to the participant, both of which were indicative of the objects used in the later tasks, with one of these objects having haptics being applied to it, while the other not. This was done to remove possible instances of unexpected output during the study, as we were not wanting to introduce accidental novelty biasing. Each participant was required to do this for both input methods before they were allowed to continue. There was no time limit on this segment, and each participant was free to interchange between input methods as many times as they desired.

Surveys were created for the different stages of the study. Pre-study surveys were provided to participants, these being used to help us understand user demographics, as well as give us an idea of prior user experience. During the study participants would be answering NASA TLX (NASA 1986) style questions after each individual segment, to understand their opinion for the varying input options and task differences. Each of these questions were on a rating scale of 1 (best) to 7 (worst). These questions were asked within simulation, with participants having full control of their choice submission. Participants answered the different questions by interacting with seven buttons, with the same interactions they had used to continue through any prior part of the study. Each and every task in the simulation was followed by this questionnaire. One final survey was provided for post-study opinions, where users were able to provide varying text-based opinions on study segments, as well as a number of extra quantitative opinion questions.

No time limits were enforced through the study, and no participant could fail too many times.

4.3.1 Questions and Hypotheses

Our study was designed to answer a number of questions, derived from the background research, our combination of independent variables, and our chosen fine motor skill tasks. We assumed a number of hypotheses based upon prior research and technological differences between the independent variables.

Q1. What are the differences in overall performance and efficiency when utilising different input methods (hand tracking vs VR controllers)?
Q2. What type of interactions and task do the different input types excel at, compared to one another?
Q3. What are the differences in overall performance and efficiency when haptics are applied to a given input method?
Q4. Do the performance statistics of the varying independent variables coincide with the overall perceived performance and opinions of the participants?

From these questions and prior information, we hypothesised that:

H1. Hand tracking should perform favourably compared to controllers when handling objects of varying size, both in overall performance, but also when it comes to accuracy of interaction.
H2. Controllers should perform favourably compared to hand tracking when handling objects over continuous, prolonged motions, such as large magnitudes of rotation.
H3. The application of haptics should provide a beneficial effect on the overall performance and perception of interactions within the tasks.

4.3.2 Study Development & Setup

We utilized the Unity game engine to create our VR simulation, with the Valve Index virtual reality headset for visual output. As the study was a comparative study between two different input methods, we chose to use the Ultraleap Stereo Infrared 170 (SIR170) for our hand tracking hardware. This would provide us with a higher field of view for the hand tracking, while also ensuring better visible range when compared to the prior Leap Motion controller (LMC) device. The SIR170 was attached to the front of the Valve Index headset. Ensuring consistency between input modalities was key to the study, so to this extent we employed the usage of Valve Index controllers for our "conventional" input method. Unlike the majority of controllers on the market, the Index controllers provide finger curl values, allowing us to closely replicate the information we would find in optical hand tracking.

Implementation of object interaction came through the usage of the Leap Motion Unity modules and interaction engine (version 4.5.1 (Ultraleap 2021b, p. 1)). The interaction engine provides a framework for dynamic object interaction, allowing for multiple sizes and shapes of objects to be touched and grabbed with hands. While it has some rudimentary support for controllers, a solution was needed to maintain the consistency in interaction between the hands and controllers. For this we implemented the HandshakeVR (Corvinus 2021), which translates SteamVR skeletal input into interaction engine hand bone positions, thus allowing the controllers to interact with all objects in a similar fashion.

Our study implemented two different methods of haptic interaction. Hand tracking utilised the Ultrahaptics STRATOS Explore ultrasound haptic array to provide mid-air haptic feedback to the user's hands, without requiring any form of contact. This device creates a "focal point" of high pressure at a desired 3D position in space through the usage of multiple ultrasound transducers, which is then moved at high speed (anywhere up to 40kHz) to create a path or pattern. This focal point movement causes displacement and micromovements of the skin, which are then felt specifically on the palmar surface of the hand. Controller input utilised linear actuators inside of the Valve Index controllers, where the user would be holding the controller and thus feeling the vibration of said motors. The linear actuators modulate several segments of a bar shaped motor forwards and backwards at varying frequencies resulting in haptic feedback.

This image has been redacted for copyright reasons.

Figure 26 The Ultrahaptics STRATOS Explore pictured on the left, and the Valve Index controllers on the right. The Ultrahaptics haptic array uses all 256 transducers to modulate the focal point. The Valve Index controllers utilise a linear actuator, positioned within the main body of the controller which is outlined in the image by the red dotted lines.

All participants were sat down in front of a standard height table (72cm) where the Ultraleap haptics array was positioned. Controllers were placed next to the array, and the user was instructed as to how to pick up and put on the controllers while inside of the VR simulation. The VR headset was worn for the entire simulation, as all workload rating questions were asked within it.

During the simulation, we recorded all object interaction, input, and head movement data within Unity. This provided the opportunity during analysis to re-review participant actions to gain further information from multiple users that may not have been apparent during testing, or to validate trends that appeared during analysis. All recorded data was anonymous, with no identifiable information saved.

We followed the Ultraleap ethics principles throughout our study. All data was fully anonymised, both for the study recordings, as well as the study surveys. All users were informed of the data that would be recorded, as well as provided an opportunity to withdraw and have their data removed at any point.

To ensure COVID-19 safety measure compliance, the study had heightened hygiene and social distancing measures. Each participant was given a VR face mask for use with the headset, this being on top of all equipment being thoroughly cleaned and sanitised between users. Study coordinators were situated behind a partition at all times.

4.3.3 Input Modalities

Throughout the study, participants were asked to use two different types of input. Each different segment of the study could ask the participant to use either of the two inputs. Participants would be instructed to either pick up or place down the VR controllers and press a continue button using the correct input method to proceed. Only the current input method would be active within the simulation, preventing the participant from continuing using the incorrect method.

Hand tracking made use of Leap Motion V4 (Orion) tracking, with the Ultraleap Stereo IR 170 (SIR170). The virtual hands would replicate every bone position of the hand with sub-centimetre accuracy. If a participant would try to grasp objects or move individual fingers, then the system would try to replicate such actions. Our visual representation of the hands consisted of a texture-less hand model, with no definition of joints or nails. All of this was done to ensure the user would focus more on the task, without having a too distinct difference between their own hands, reducing possible inaccuracy distractions.

While we could have opted to use other hand tracking technology, such as that of Facebook's Oculus Quest (Oculus 2019a) or Google's MediaPipe (Zhang et al. 2020), most current implementations have distinct limitations that would hinder our study development. First, many of the current hand tracking implementations are platform limited, either requiring a specific computing platform or operating system to work. Non-Windows based platforms would not align with any of the other technology or tools that were going to be used, therefore ruling out several options. Secondly, platforms that may include hand tracking technology often did not include favourable conventional controller options to help contrast against the hand tracking. In scenarios where they did, the interchange between said devices was not technically feasible at the time of study development. Finally, many of the available hand tracking platforms do not adjust for spatial congruity, with hands being tracked, but not understood in regards to their size or relative positioning in 3D space. This information is critical when working beyond simple gesture recognition, as 3D objects cannot be reliably manipulated without understanding where they are relative to the camera.

Overall optical hand tracking interactions with objects were controlled through a multitude of bone information, provided via the usage of the Leap Motion interaction engine. The point at which objects were able to be successfully grabbed was calculated by the collision points between at least two fingers of the hand and the desired object. Once objects were shown to be "grasped" they were then attached to the overall transformation space of the hand, moving with it. This space was taken from an average of the grasped fingers and palm, which may change during movement due to changes in finger curl, thus updating in real time with the participant hand movements.

VR controllers' digital representations were created in a way to mimic that of the hand tracking. The Valve Index controllers implement finger curl tracking into their grip, allowing for fingers to be moved in a similar fashion to if they were being tracked one to one. By this extension we were able to recreate a set of hands that were of similar visual representation to the hand tracked versions, while also physically operating in a similar way. All finger bones provided collision data, just like the hand tracked counterpart, and could affect the overall positions of objects.

Figure 27 Two images showing the differences between the physical and virtual representation of the hand. The left image showing how the participant would be holding the VR controller, and the right showing the resulting hand pose.

Unlike with hand tracking, object grasping with controllers was a more binary affair, with fewer degrees of freedom. The overall lack of omni-directional movement meant that finger positions would only translate to their real world relative position on the controller, rather than in full 3D space. This meant that it was impossible for the thumb, which was tracked by the upper touch pad, to be used in conjunction with only the little finger. Due to this, grasping was not as dynamic compared to hand tracking, being limited to two possible hand poses. Any combination of the thumb and fingers, simply the last three fingers of the hand, or bringing all fingers into a fist pose, could be used to grasp and pickup objects. Once an object was grabbed, its translations would match the controller, relative to when it was grabbed, with no change based on the position of fingers. This movement would continue until none of the possible hand poses were active.

Both input methods allowed for grasp poses that could change over the course of object interaction. As long as one of the possible variables deciding if the object was grasped was still met, then the object would remain grasped. This would allow participants to change and adjust their poses to better suit the resulting desired output of their actions.

4.3.4 Tasks

Our study utilised two different tasks, both making use of uniform digital blocks as their interaction objects. These blocks were simple in visual design, being white with smooth black edges, designed as such to help visual clarity of the object. Neither task had a time limit on it, nor could the participant cause too many failures and end the task prematurely. All participants undertook the same set of tasks and input/haptic combinations, however, the order in which they received the tasks, the input to be used, and the inclusion of haptics in the tasks, was entirely randomised. As there were a total of two tasks, two inputs, and two haptics options, with the participant repeating each task twice, there were 16 total individual tasks that the participant would undertake.

Before the beginning of each task, the participants had to confirm their continuation by pressing a continue button using the specified input method. The randomised order meant that participants would often be repeatedly picking up and putting down the controllers between tasks.

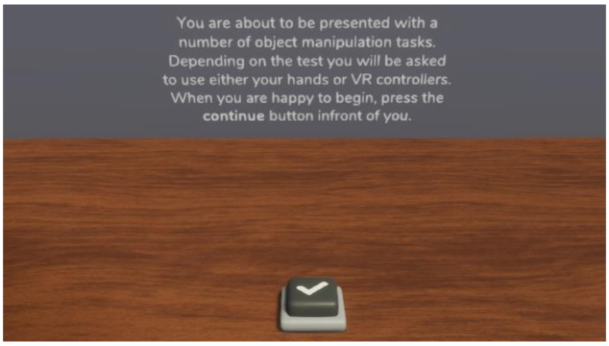

You are about to be presented with a
number of object manipulation tasks.
Depending on the test you will be asked
to use either your hands or VR controllers.
When you are happy to begin, press the
continue button infront of you.

Figure 28 The button that the participant would use to confirm their choices throughout the study. This was often tied to specific input combinations, requiring either the optical or controller based hands to be used based on the following task.

Success and failure events were reported through sound effects, with a positive "ding" effect for success and a negative "dong" effect for failure. These were the only audio cues in the study, beyond a click sound when the continue button was pressed.

Between tasks participants were asked to interact with a small survey, which was designed to rate their opinions of their performance. This survey was based on the NASA TLX system, with participants directly picking their answers using the current input method.

4.3.4.1 Block Stacking

The block stacking task was designed to test the effectiveness of the input method, when asked to manipulate varying sized objects with precision and care. Due to the nature of the task, it would require people to make focused movements, with direct intent of their actions. Information on the task was administered through text descriptions within the simulation, which would update as necessary throughout.

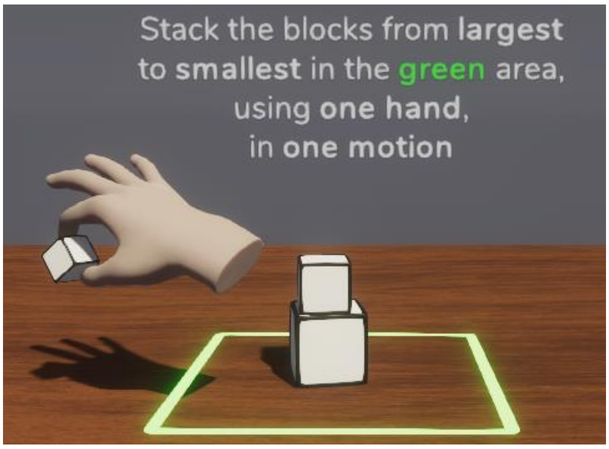

Stack the blocks from **largest**
to **smallest** in the **green** area,
using **one hand**,
in **one motion**

Figure 29 The block stacking task's first stage, where the participant has already stacked two blocks and is about to stack the third. The block on the left would always appear in the same position, but with randomised rotations.

There were two halves to the task, completed in order. The first part being the stacking portion of the task. Participants would receive three cube blocks in reducing size and be asked to stack them from largest to smallest, within a specific area on a virtual table. Block sizes were reported based on their longest edge, with diameters of 0.07cm (large), 0.05cm (medium), and 0.03cm (small). Each block would be presented in order, in the same mid-air position, with a randomised rotation. Blocks would be revealed one after another once they were successfully placed.

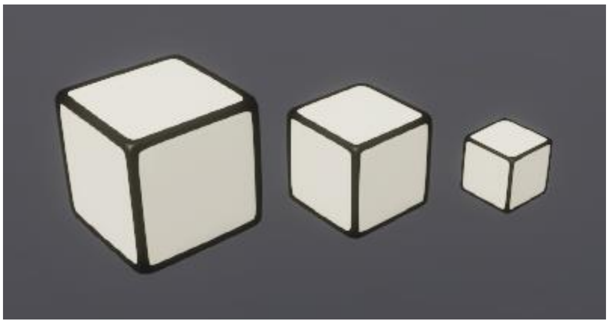
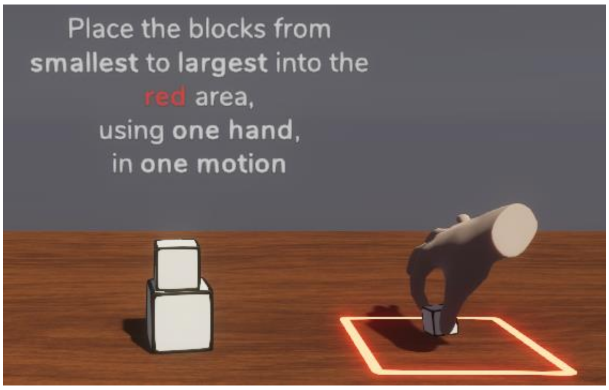


Figure 30 The cubes used within the study, presented from largest to smallest. These cubes were used for both the stacking and object sorting task. The middle cube was used for the object sorting task.

The second half of the task consisted of a dismantling process. Participants would be required to dismantle the blocks they had just stacked, from smallest to largest, into a specific designated area. Blocks would be in the exact positions that they were placed during the stacking stage, meaning that difficulty of dismantling can often be increased based on prior placement by the participant. Once all blocks were successfully removed, the task was complete.



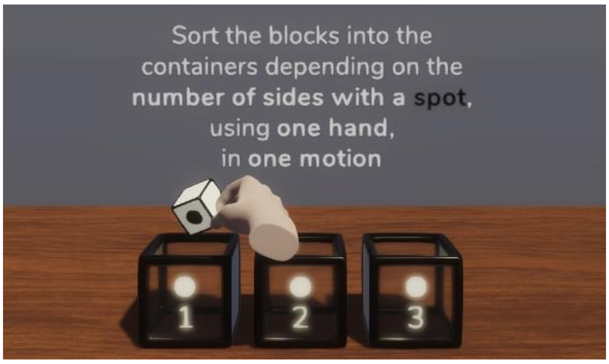
Place the blocks from
smallest to **largest** into the
red area,
using **one hand**,
in **one motion**

Figure 31 The block stacking task's dismantling stage, where the participant is removing the first of the three blocks into the designated area.

Throughout the task, participants were only permitted to grasp blocks with one hand, and then place them in the correct position in one motion without letting go. If the user failed to do so, then it would simply reset to the previous successful state. This included returning to the original rotation value in the stacking stage, a decision that forces the user to adapt their hand pose. Failure and success states were only calculated after the participant had released a block. This was performed once every second and registered when the positional and rotational velocity of the object had reached a sufficiently low threshold. In this case it was speeds less than 0.06m/s. Other failure states could occur when the user knocked blocks beyond a reasonable threshold, or when users tried to grasp the incorrect block during the dismantling portion.

4.3.4.2 Object Sorting

The object sorting task was designed to test the effectiveness of the input method, when asked to manipulate objects in a continuous hand pose and motion with consistency. Each participant had to sort six blocks into three containers on the virtual table in front of them, based on the information presented on the blocks. The goal of the task forced participants to make large, highly explorative movements of objects, with high amounts of rotational movement. Instructions for the task were administered through text descriptions within the simulation, which would update based on progress throughout the task.



Sort the blocks into the containers depending on the number of sides with a spot, using one hand, in one motion

Figure 32 A participant sorting a block into a container during the object sorting task.

The object sorting task was undertaken as one continuous task, where users would be viewing information on a block and sorting it into the respective container matching the information. This process had to be completed in one continuous grasping movement, using one hand, without letting go. Each block was presented in mid-air with a randomised rotation. These blocks were the same base visual as found within the block stacking task, with the same size as the medium block (0.05cmlongest edge). Differences were made by adding an extra texture to each block that added either one, two, or three total spots to varying sides of the block. There was a total of six blocks to be organised, with two blocks per container, and five possible varying spots textures. Due to the nature of the possible patterns available it would result in one option for one spot, which was repeated twice, two possible options for two spots, and two possibilities for three spots, thus resulting in five spot combinations (one for one spot, two for two spots, and two for three). Order of these blocks was randomised, along with rotations on initial spawn.

When a block was organised into the correct container it would be removed from the scene, with a confirmatory "ding" sound. Objects were not retained within their containers as doing so would have presented the participant with information regarding task progress and thus given later blocks in the task completion bias. If a block was placed into an incorrect container, or that it was dropped or let go of prematurely then it would be classified as a failure. Success and failure tests were performed when block movement and rotational velocity was under a threshold, allowing for dropping or throwing methods of completion.

4.3.5 Haptics

A key component of the study was the implementation of haptics at different stages. Both input methods and tasks had haptics applied during the study, with 50% of combinations having haptics.

When haptics were applied to the simulation, it was done so in a uniform method across both input methods and interactions. Each interactable object within the study, be it a button or block, would cause haptic feedback to be triggered. This would occur when the participant would come into contact with the interactable objects, where any part of the hand collision would cause haptics to be fired. Haptics would be continuously applied until the hand was no longer in contact with any object. Both haptic effects were implemented in a way where the results would be similar, with a consistent, relatively strong, feedback force.

While the devices used had considerably large differences between them, with one being contact based haptics and one not, the final effects used were designed to maximise their outputs. This was achieved by operating both devices at their maximum possible effective output. Respectively, this was at an average of 155dB of pressure over one second for the ultrasound array, while an intensity of one for the VR controller haptics.

4.3.5.1 Mid-Air Haptic Implementation

Optical hand tracking specific haptics were implemented through the usage of the Ultraleap STRATOS Explore haptic array. Our implementation was designed to provide as high percentage of haptic feedback being applied to the participants hand as possible, without having to rely on particularly cumbersome or restrictive setups. This meant only relying on a single hand tracking device to report positions, but also not affixing the hand tracker to the haptic array itself. Both of those setups require extra costs, be it increased performance load on the system and extra hardware costs (when using two devices), or physical restraints to what and where the simulation could be interacted with. To this extent, we chose to develop a method of implementation that does not rely on the direct position of the hand for placing haptics, but that of the currently interacting objects.

The haptic implementation we developed relies on the interaction between the virtual hand and the virtual objects. As the objects in the environment are touched by the participants hands, and therefore intersected, they become "active" within the implementation and are then processed to produce haptic effects. Sensations are applied on an object level, instead of relying on bone positions reported by the hand.

The focal point that the haptic array creates is around 8mm in diameter (Ito et al. 2016), which is significantly smaller than any of the objects we were using within our study. Producing the haptic effect using the entire area of the object would be inefficient as we could not be assured as to where the hand was touching. To alleviate this, each interactable object was subdivided into regions, based upon the overall size of the mesh. Similar previous implementations rely on theories such as voxels (Martinez et al. 2019), being uniform in size across all three axes of each division. Unlike voxels which are completely uniform in size, our method used regions that were sized based upon a pre-defined 3 axis vector. This vector defined how many subdivisions would occur across each of the object's axes. These regions were defined before runtime, however, could be adjusted or recalculated in real time if necessary. All regions would follow the overall position, scale, and rotation of the object, allowing for easy recalculation based upon object movement.

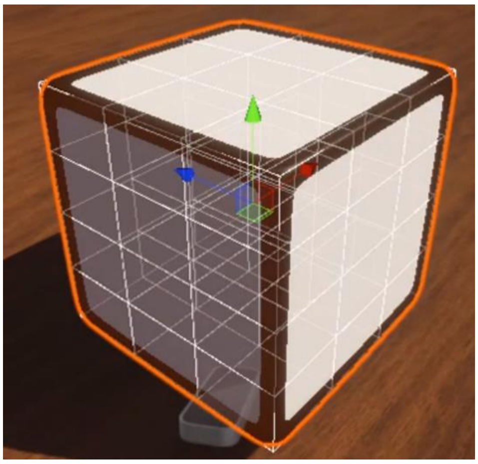


Figure 33 An object showing the subdivisions calculated. In this example the developer has defined a subdivision vector of 3x4x3.

To ensure that the hand would be receiving haptics, even when not receiving optimal positioning from the tracking, we expanded the overall radius of the finger bones by 25%. This would mean the fingers would still report intersections with objects, even if there was a slight difference between the real-world position of the hands due to tracking jitter or inaccuracies.

Intersection tests would be run on the interactable object as a two-stage process. If the object is in its idle/non-contact stage, then only the top-level intersection test to check whether a hand had touched it. Once a hand had started intersecting with the object then the second level, region intersection test would be fired. All regions that were intersecting the hands would be collated, including those that are from other objects, into one final region.

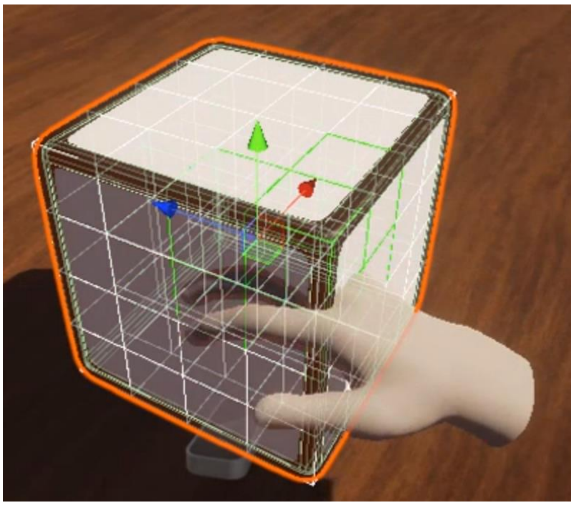


Figure 34 Shows the object being intersected by the hand. Regions being intersected by the hand are shown with a green outline.

This collated region would be used as the volume in which possible haptic positions would be calculated. These haptic positions were generated as a random array of points inside of the region. Every few milliseconds, the current index of the array would be changed, with the haptic sensation moving its position to the next position. Both the total number of possible positions, and the speed of iteration was developer defined, in our case we chose to use 40 points per individual region and an iteration speed of 40 milliseconds. These values gave us a good balance between consistently filling the volume with haptic feedback, thus increasing hit rate with the hand, while also retaining a high level of object presence.

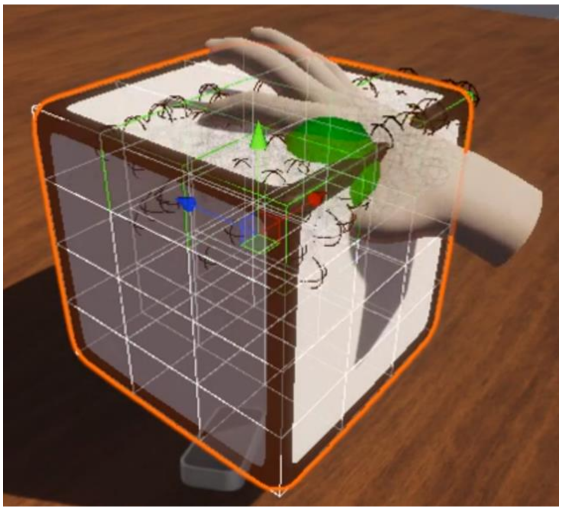


Figure 35 Shows the object being intersected and the resulting haptic positions. The currently intersected regions are outlined green, the possible locations of the haptics are designated by black sphere outlines, and the current position of the haptics as a solid green sphere.

Combined with the positions, a secondary motion could be applied on top of this, for our situation we chose to use a circular motion as this helped to reduce the overall sound of the sensation. This circular motion was modulated at a frequency of 60Hz. Each instance of the circle had a radius equal to that of the average size of all currently interacting haptic regions. In practice this would mean that a region with a size of 0.2cm, and another of 0.3cm would result in a circle radius of 0.25cm. This change in size was performed to ensure that the currently generated haptics were respective of the currently interacting regions of the shape, without to directly specify the individual sizes.

There were two conditions that caused positions to be regenerated. Either that the collection of intersecting regions had changed, or that the overall collection of positions had been fully iterated over. Once either of these conditions had been met, then the generation of points would re-occur. Counter to this, the intersection tests would occur every physics step within the simulation, as long as they were being intersected with the object as a whole.

This solution improved the successful hit rate of providing the haptics onto the hand, as the focal point can often be slightly misaligned when targeting the pure hand tracking position. Variable positioning from the headset, and therefore hand tracking origin, and jitter introduced by the SteamVR tracking solution, did not reduce the overall effect of the implementation as the object regions were sufficiently larger than the hand.

Our solution for mid-air haptics was designed to help produce beneficial results when working with a relatively common scenario, where the hand tracking device is attached to the VR headset. This implementation meant that participants would still be able to visually see their hands at all times. The current standard implementation of mid-air haptics relies almost entirely on a fixed origin tracker, which is generally attached to the mid-air haptic array. While the standard implementation provides a more accurate set of hands for applying haptics due to lack of secondary tracking jitter, it severely limits the area in which the user can interact with the system. Hand tracking provided from a headset mounted orientation was proven to produce more stable results, compared to a desk mounted orientation, even if less accurate overall for the application of haptics. During prior studies and work, we had witnessed that desktop mounted hand tracking through Leap Motion hand trackers provides poorer results due to it using an older tracking software stack, older hardware with reduced field of view, and multiple issues with hand occlusion. At times these issues can cause in descript objects in the trackers field of view to be reported as hands, thus causing the participant to become distracted due to unnatural or unexpected movements. Ensuring a high level of visual consistency throughout the study was paramount, thus why we implemented this method of haptic application, making full use of the newer technology stacks and tracking hardware available.

4.3.5.2 Controller Haptic Implementation

The controller haptic implementation was a decidedly simpler approach, utilising the SteamVR haptics methods to generate them. As the controllers were bound to the participants hands and could not generate haptics beyond themselves, there was no need to include information about positions within objects. We simply relied on Unity's physics collisions, if the hands came into contact with an interactable object then we initiated the haptic feedback at a frequency of 100Hz with full intensity. These values were chosen as they were similar in feeling to the mid-air haptic solution, albeit stronger in overall force. Once there were no more objects touching the hand, we then disabled the haptics.

4.3.6 Data Recording

Throughout the study we recorded every single movement that the participant was making, while they were performing the two tasks. This was achieved by using PlayRecorder (Clark 2021), a Unity logic recorder we had developed. This was built directly into the simulation, allowing it to record underlying logic, rather than just still video frames like a traditional video recorder. Similarly in nature to video compression codecs, PlayRecorder only records changes between objects but instead relies entirely on the game engine to perform rendering and visuals for playback.

PlayRecorder was used to record transform data (position, rotation, scale) for all task blocks, hands, and head position, state changes for tasks, and post task survey questions. This information was taken from the built application, and then stored into binary files that were then later loaded back into Unity for analysis. On top of this raw information, the recording files stored simple text-based messages combined with a corresponding timestamp. These messages act like keyframes or events, and were entirely customisable, with any component of the recording being able to record messages. We used this to record when objects were being grasped or released, and tasks were progressed or failed. All our recordings were performed at up to 60 times per second.

All of the information was completely anonymised, as it was only being recorded from within the simulation. Although participant movements are unique, there was nothing within the recording that could be used to identify the participant.

4.3.7 Study Protocol

The following segment covers the protocols and surrounding information used to govern the study before, during, and after a participant undertook it.

4.3.7.1 Study Surveys

All participants were required to complete two different surveys during our study. One survey was conducted prior to the simulation, with the second survey being completed after the participant had completed it.

The pre-simulation survey was completed up to one week prior to the participants study time slot. This survey was designed to gain an insight into the demographical information of our participants, primarily probing into prior usage of technology. We asked for the participants age, gender, VR/AR usage, and their other technology device usage. Questions also included targeting information regarding their prior haptic usage, specifically to know about any preconceived expectations people may have. Most of this data was used to help understand possible biasing and discerning the overall technical level of our participants. While parts of this information were not technically integral to the overall study, it was useful when discovering trends within data analysis.

The secondary, post-study survey was completed shortly after the participant had completed the VR simulation portion. Participants would be questioned about their opinions, specifically around the individual tasks, the input methods, haptic effects to different study events, and the overall study as a whole. Participants were also free to express any comments about the different segments of the study. As to keep conformity with the study, quantitative questions (where applicable) were asked in a 7-point Likert rating format, the same as found within the NASA TLX questions.

Both surveys were conducted outside of the simulation, and in the participants own time. We collected the results of the surveys through Google Forms. The pre study and post study survey questions can be seen in the appendix (7.1.2.1 Pre Study, 7.1.2.2 Post Study).

4.3.7.2 Participants

We recruited 17 participants (11 male, 6 female) to take part in our study, including 14 Ultraleap employees, and 3 external participants. Each participant had no prior knowledge of the tasks to be completed. All participants signed full consent disclosure to take part in the study and were fully aware that their simulation movements and choices were being recorded.

Our participants were split relatively equally in prior VR usage, with 5 participants regularly using VR, 6 rarely using, and 6 either being new to or using VR just once. 5 of our participants had used the Valve Index before, and thus had prior experience with using the controllers.

There was a slightly higher skew of participants with prior hand tracking experience, with 7 regularly using, 5 rarely using, and 5 with no prior experience. We queried participants on the context of their interactions with hand tracking, where they had to place how likely their experiences interacting with hand tracking was in either a desktop (indirect/third person) or VR (direct/first person) scenario. This was queried across a scale of 0 (desktop) to 10 (VR), with a resulting geometric mean of 1.784, showing a strong skew towards desktop scenarios. We could summarise that our participants were likely to be relatively familiar with hand tracking, although not within the context we were testing within.

We queried the usage of ultrasound haptics, and once again the skew of experience was higher, with 11 regularly using, 2 rarely using, and 4 with no prior experience.

From our questions we could assume that our participants as a whole were experienced with mid-air haptics, and hand tracking, and partly experienced with VR technologies.

4.3.7.3 COVID-19 Impact on Testing

Due to the nature of when testing occurred, during the middle of the COVID-19 pandemic, it was not entirely feasible to recruit large numbers of participants from outside of our workplace environment. This was combined with the key nature of the study, where a participant would not only be physically touching and holding controllers, but would also be wearing a head mounted VR device for the entirety of the study.

We were unable to recruit further participants or run a particularly large amount of initial preliminary people due to government guidance and the re-introduction of COVID-19 related lockdowns around November 2020 in the UK. While the circumstances of the testing period should not affect the results, the smaller sample size will unfortunately play a role. To account for this, we repeated each task with every user twice to help ensure we were limiting biasing.

4.3.7.4 Statistics Analysis

Our data analysis was a two-stage process, first involving extracting and understanding data from our Unity simulation recordings, and then secondly processing said information and their accompanying surveys. All data was processed within Jupyter Lab for Python, with the numpy, pandas, matplotlib, and seaborn packages. This selection of software was used to analyse the data, as well as create graphs and visualisations.

4.3.7.5 Data Recordings

Throughout the study, participant playthroughs were recorded using our custom developed analytics tool called PlayRecorder. These files included all the participants' physical and object movements for all input and haptic combinations. With this information we created a post study analysis environment within the Unity game engine that allowed us to collect varying levels of object analytics.

Data was collected both manually and automatically from the system. Manual data primarily consisted of information regarding specific niche failure states, unexpected successful completions. Automatic data was collected for individual blocks throughout the study, with each block having a number of different statistics recorded. Where not specified, these statistics were recorded for the block that was successfully actioned in the task, and not the prior attempts. As an example, if the participant failed three times on the second block of the stacking task, then only the fourth block's statistics would be registered. While registering all failures statistics could have been possible, it would have resulted in varying statistic counts, and thus unnecessarily complicating analysis.

Statistic	**Data**
Failures until the current block was successfully actioned	Numeric count (0..N)
Hand pose of grasp	Pose (pinch, multi-finger, whole hand)
Duration of grasp	Time (seconds)
Duration between hand entering a radius of 10cm of the object and successfully grasping the object (grasp time)	Time (seconds)
Duration between releasing the object and hand leaving a 10cm radius of the object (release time)	Time (seconds)
Total positional movement of the object during grasp	Distance in meters (0..N)
Total rotational movement of the object during grasp	Rotation in degrees (0..N)
Positional difference between centroids of large and medium sized block (within the block stacking task)	Distance in meters (0..N)
Positional difference between centroids of medium and small sized block (within the block stacking task)	Distance in meters (0..N)

Table 6 The above table outlines the different statistics and data types we recorded and collected from our data recordings.

The recorded statistics are outlined in Table 6. They were chosen as they should provide a good insight into possible trends between the input and haptic conditions throughout the study. Several of these statistics were reported in other similar studies, such as grasp and release times, where they proved insightful to the final conclusions (Masurovsky et al. 2020). Nearly all of the statistics were generic enough to be applicable to both tasks.

4.3.7.6 Data Disclosure

No specific participant's data was removed during the data analysis, with all task recordings and surveys being used. Due to the nature of how the tests were structured, each block interaction was stored as a single data point. A total of 35 individual tests were completed, with the first participant only receiving one set of test combinations. This singular set of data did not change any significance tests and was therefore kept in during analysis. Once all data sets were collected and sanitised, we were left with 808 individual block interactions for the block stacking task, and 840 for the object sorting task.

In a small segment of tests, participants encountered two different bugs during the block stacking task. This bug caused the dismantling stage of the task to skip one block. This would cause the task to finish prematurely, and in cases where this occurred the completion time of the dismantling stage of task was modified to be 150%. The other bug would cause a small number of blocks to erratically move due to physics collisions, where this was occurred, failure rates were adjusted to account for these unintended effects. These bugs were the reason for discrepancies between overall data set count.

4.4 Results

The results are explained in the following sections, where we will divulge how participants performance aligned with their opinions, both in the simulation and in the surrounding surveys. Data will be subdivided between the two different tasks, with all cases being specified if they are not. Block stacking (BS) featured a number of dependent variables: overall task time, stacking time, dismantling time, placement accuracy between block sizes, number of failures, block grasp & release times, mean movement time, and hand pose type during grasp. In addition, the NASA-Task Load Index (TLX) questionnaire was presented between conditions in order to measure overall workload during each condition, with the individual factors of Mental Demand, Physical Demand, Temporal Demand, Effort, Performance, Frustration being assessed. Object sorting (OS) featured the same dependent variables except for placement accuracy, as this metric was specific to the BS task only, and stacking and dismantling times, as the OS task was completed as one continuous stage. Block size, or a corresponding similar variable, was not applicable to OS as each of the blocks were of uniform size, being the same size as the BS medium block, and required the same level of interaction throughout the task. Each reference to block size corresponds to the largest edge diameter of the block, where the large block is of 0.07cm, medium of 0.05cm, and small of 0.03cm. Although not directly a recorded statistic, no participant swapped between input methods during the acclimatisation stage more than once.

Multiple dependent and independent variable factors were shown to be heavily skewed or not normally distributed meaning we were not able to perform standard parametric analysis. Due to this distribution of the results, and coinciding dependent variables, we applied an Aligned Rank Transform (ART) to our continuous and ordinal variables for both tasks. These were then analysed using factorial non-parametric repeated measures analysis of variance (ANOVA) tests. This transformation and subsequent ANOVA testing allowed us to analyse the differences in interactions between independents and the overall main effects. In scenarios where significant interactions between factors were observed, we then proceeded to perform follow-up pairwise T-Tests to find the individual significant pairings. Geometric means (gM), geometric standard deviation (gSD), medians (Mdn), interquartile range (IQR), lower quartile (LQ), and upper quartile (UQ) will be reported for each significant due to the data not being normally distributed. Geometric values are reported due to the data being transformed by the ART. All dependent variables will have their values reported, alongside their data skews.

Our study had two dependent variables of failure rate and hand pose type, which required a different method of analysis to our continuous and ordinal values. Failure rates were count data, while hand pose type was multinomial data. These were both analysed using Generalized Estimating Equations (GEE). In the case of failure rates, data for each condition was assumed to follow a negative binomial probability distribution due to having identified that each distribution had a variance substantially larger than its mean, violating a required assumption for a Poisson-based regression. Hand poses were assumed to follow an ordinal logistic probability due to being categorical. In the case of ordinal logistic analysis, we will be reporting the parameter estimations which summarised the effect of each predictor through their coefficients. This reports the likelihood of a value in the group being a multiple higher than an another. With the example of hand poses, this equates to higher values being more likely to be a whole handed grasp, while lower values being closer to a pinch.

Graphs will be displayed for significant pairwise interactions. For continuous data, these will be presented as medians with lower and upper quantiles being displayed, while the ordinal data will be presented using the geometric means and geometric standard deviations.

4.4.1 Block Stacking Task

4.4.1.1 Overall Task Completion Time

The time, in seconds, to complete the entire task was recorded. Timing commenced when the participant pressed the continue button to start the task, and ended when the final block was removed from the stack.

Overall task completion time values were *(gM = 43.422, gSD = 1.551, Mdn = 42.11, IQR = 27.344, LQ = 30.217, UQ = 57.561)*, and the distribution of data was heavily positively skewed *(3.298)*.

Analysis was conducted to assess the effect of haptics, input type, and task repetition on the overall task completion time. No statistically significant 3-way interaction effect was found. Completion time differed based on a 2-way interaction between trial repetitions and input type *F (1, 16) = 6.599, p = 0.021*. Follow-up pairwise T-Test comparisons showed that controllers were slower *(gM = 47.224, gSD = 1.598, Mdn = 48.052, IQR = 35.425, LQ = 30.967, UQ = 66.392)* than hands *(gM = 42.143, gSD = 1.423, Mdn = 41.933, IQR = 19.89, LQ = 32.081, UQ = 51.971)* in the first trial, however this was not the case for the second trial. A main effect of haptic application *F (1, 16) = 5.740, p = 0.029* was reported, with haptic application being slower *(gM = 45.181, gSD = 1.584, Mdn = 44.146, IQR = 30.771, LQ = 30.208, UQ = 60.979)* than without *(gM = 41.73, gSD = 1.517, Mdn = 37.517, IQR = 24.645, LQ = 31.054, UQ = 55.699)*. No further significant interactions were reported.

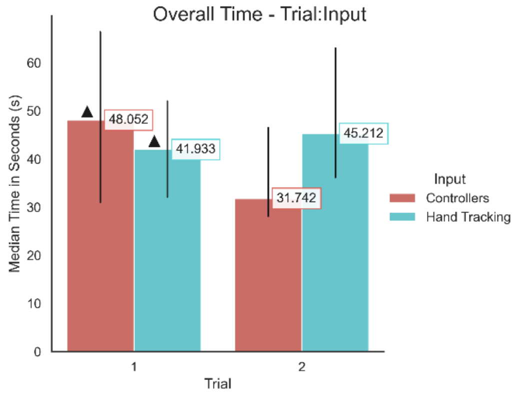


Figure 36 Graph showing the median, lower quantile, and upper quantile of overall completion time for the BS task, with pairwise comparison grouped by trial repetition and input method. ▲ denotes significant pairwise interaction.

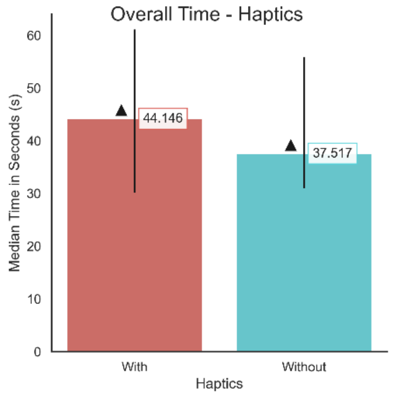


Figure 37 Graph showing the median, lower quantile, and upper quantile of overall completion time for the BS task, with pairwise comparison grouped by haptic application. ▲ denotes significant pairwise interaction.

4.4.1.2 Stacking and Dismantling Time

The time, in seconds, to complete the stacking and dismantling stages of the BS task were individually recorded. Timing for the stacking started when participants pressed the continue button and then stopped when the participant placed the last block of the tower. Dismantling times started once the participant started the corresponding part of the task, and was finalised once the last block had been successfully removed.

Stacking times were *(gM = 21.59, gSD = 1.615, Mdn = 19.75, IQR = 15.392, LQ = 14.971, UQ = 30.362)*, and the distribution was positively skewed *(2.503)*. Dismantling times were *(gM = 19.675, gSD = 1.751, Mdn = 18.742, IQR = 12.567, LQ = 12.746, UQ = 25.313)* and was heavily positively skewed *(4.908)*.

We analysed the effect of haptics, input type, and task repetition on the two time parameters independently. The stacking time reported a single significant 2-way interaction between the application of haptics and the trial *F (1, 16) = 10.292, p = 0.005*. Resulting pairwise T-Tests reported that the addition of haptics resulted in slower *(without haptics, gM = 19.347, gSD = 1.785, Mdn = 17.333, IQR = 9.988, LQ = 14.508, UQ = 24.496) (with haptics, gM = 23.207, gSD = 1.58, Mdn = 23.633, IQR = 18.208, LQ = 14.987, UQ = 33.196)* times during the second trial, but not during the first. No further significant interactions were reported for the stacking times.

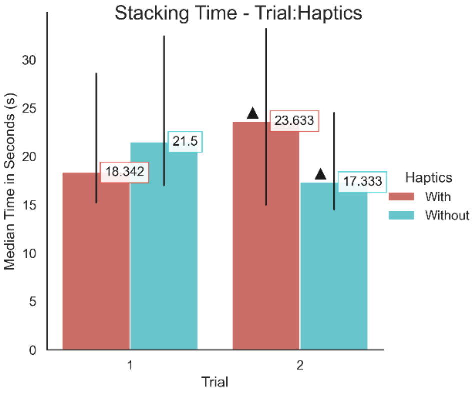


Figure 38 Graph showing the median, lower quantile, and upper quantile of completion time of the stacking element of the BS task, with pairwise comparison grouped by trial repetition and haptic application. ▲ denotes significant pairwise interaction.

Dismantling reported two 2-way significant interactions between trial and input type *F (1, 16) = 8.140, p = 0.012*, and trial and haptics *F (1, 16) = 5.636, p = 0.030*. Pairwise T-Tests showed controllers being slower *(gM = 22.494, gSD = 1.895, Mdn = 20.525, IQR = 27.962, LQ = 12.971, UQ = 40.933)* than hand tracking *(gM = 18.64, gSD = 1.638, Mdn = 16.762, IQR = 8.943, LQ = 13.367, UQ = 22.309)* during the first trial, however, it did not report any significant interaction for the second trial. While the trial and haptic ANOVA reported significance, the corresponding pairwise t-tests did not. A significant interaction was reported for the haptic main effect during dismantling *F (1, 16) = 5.558, p = 0.031*, where haptics were also slower *(gM = 21.034, gSD = 1.867, Mdn = 19.658, IQR = 14.708, LQ = 12.579, UQ = 27.288)* than without *(gM = 18.405, gSD = 1.623, Mdn = 18.225, IQR = 11.013, LQ = 12.821, UQ = 23.833)*. No other main effect was shown to be of significant interaction.

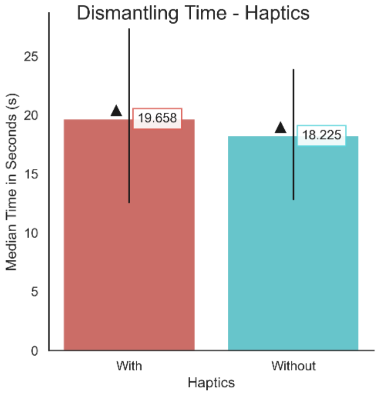


Figure 39 Graph showing the median, lower quantile, and upper quantile of completion time of the dismantling element of the BS task, with pairwise comparison grouped by haptics. ▲ denotes significant pairwise interaction.

4.4.1.3 Object Grasp Time

Grasp time was recorded, in seconds, as the time between the participants' input method entering a radius of 10cm of the extent of the object and the point of successfully grasping the object. Each object was floating in mid-air at a randomised rotation. For example, if the object had a width of 4cm then the time would be started when the participant entered a radius of 12cm. This was primarily recorded to understand the differences that the input type have on the speed of interaction with objects.

Grasp times as a whole were *(gM = 0.41, gSD = 2.441, Mdn = 0.383, IQR = 0.383, LQ = 0.25, UQ = 0.633)*, with a heavy positive skew *(5.256)*.

Test	df	F	p
Trial	(1, 16)	0.934	0.348
Haptics	(1, 16)	2.279	0.151
Trial:Haptics	(1, 16)	6.884	0.018
Input	(1, 16)	44.735	<0.001
Trial:Input	(1, 16)	13.741	0.002
Haptics:Input	(1, 16)	4.314	0.054
Trial:Haptics:Input	(1, 16)	3.121	0.096
Block	(2, 32)	5.966	0.006
Trial:Block	(2, 32)	9.183	0.001
Haptics:Block	(2, 32)	11.127	<0.001
Trial:Haptics:Block	(2, 32)	1.6	0.218
Input:Block	(2, 32)	4.165	0.025
Trial:Input:Block	(2, 32)	9.783	<0.001
Haptics:Input:Block	(2, 32)	15.36	<0.001
Trial:Haptics:Input:Block	(2, 32)	2.991	0.064

Table 7 Table showing the results of the ANOVA tests for the grasp times. Green highlights denote significant interactions.

We compared the effect of haptics, input type, block size, and task repetition. Results of the repeated-measures ANOVA can be seen in Table 7. A main effect of stage, whether the statistical value had been reported during the stacking or dismantling portion of the task, had multiple significant effects and provided a large quantity of pairwise T-Test results and was thus tested independently *F (1, 16) = 39.541, p = <0.001,* showing faster grasp times throughout the stacking stage *(during stacking gM = 0.357, gSD = 2.56, Mdn = 0.333, IQR = 0.333, LQ = 0.217, UQ = 0.55) (during dismantling gM = 0.471, gSD = 2.27, Mdn = 0.433, IQR = 0.388, LQ = 0.3, UQ = 0.688)*.

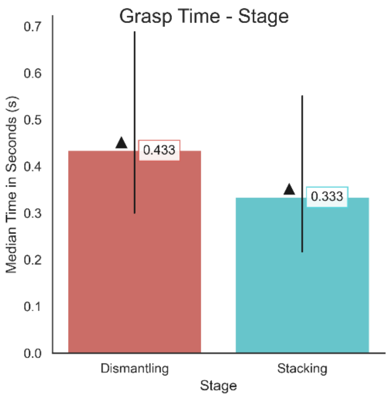


Figure 40 Graph showing the median, lower quantile, and upper quantile of grasp times for the BS task, with pairwise comparison grouped by stage of task. ▲ denotes significant pairwise interaction.

Main effect of input showed faster grasp times using hands *(gM = 0.318, gSD = 2.394, Mdn = 0.317, IQR = 0.25, LQ = 0.217, UQ = 0.467)* compared to controllers *(gM = 0.529, gSD = 2.313, Mdn = 0.483, IQR = 0.492, LQ = 0.329, UQ = 0.821)*. This was also found in a 2-way effect between trials and input, where the hands were faster in the second trial *(gM = 0.311, gSD = 2.34, Mdn = 0.317, IQR = 0.283, LQ = 0.2, UQ = 0.483)* than the first *(gM = 0.325, gSD = 2.451, Mdn = 0.317, IQR = 0.238, LQ = 0.217, UQ = 0.454)*, as well as being faster than controllers during the second trial *(gM = 0.485, gSD = 2.281, Mdn = 0.447, IQR = 0.421, LQ = 0.312, UQ = 0.733)*.

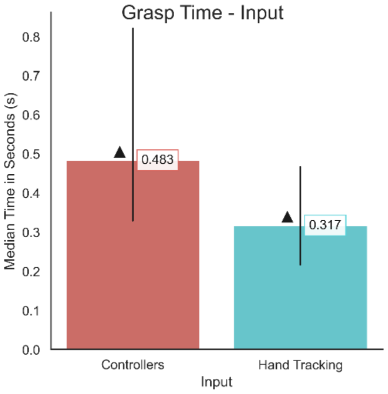


Figure 41 Graph showing the median, lower quantile, and upper quantile of grasp times for the BS task, with pairwise comparison grouped by input type. ▲ denotes significant pairwise interaction.

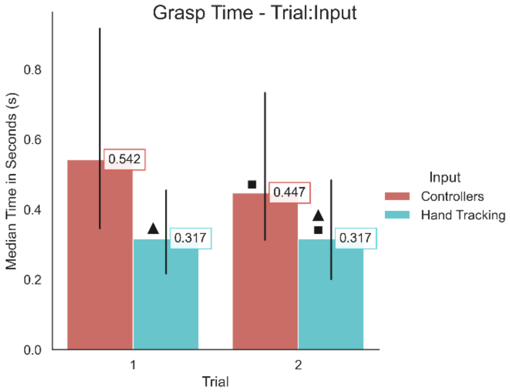


Figure 42 Graph showing the median, lower quantile, and upper quantile of grasp times for the BS task, with pairwise comparison grouped by trial repetition and input type. ▲■ denotes significant pairwise interactions.

Haptics were faster during the second trial *(gM = 0.396, gSD = 2.591, Mdn = 0.383, IQR = 0.404, LQ = 0.233, UQ = 0.637)* than the first trial *(gM = 0.427, gSD = 2.574, Mdn = 0.392, IQR = 0.421, LQ = 0.25, UQ = 0.671)*, while also being faster than the first trial without haptics *(gM = 0.438, gSD = 2.431, Mdn = 0.387, IQR = 0.367, LQ = 0.283, UQ = 0.65)*, however there was no significant interaction reported for the haptic main effect.

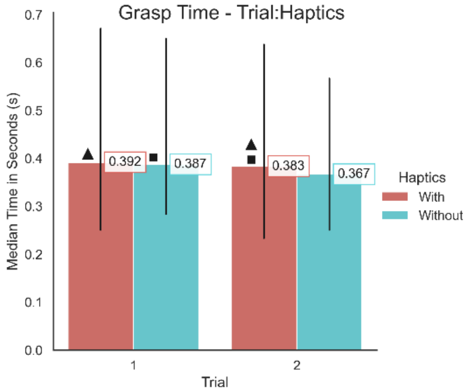


Figure 43 Graph showing the median, lower quantile, and upper quantile of grasp times for the BS task, with pairwise comparison grouped by trial repetition and haptic application. ▲■ denotes significant pairwise interactions.

Block sizes showed differences between the small *(gM = 0.404, gSD = 2.5, Mdn = 0.395, IQR = 0.388, LQ = 0.25, UQ = 0.638)* and medium *(gM = 0.387, gSD = 2.326, Mdn = 0.383, IQR = 0.333, LQ = 0.25, UQ = 0.583)* block, and medium and large *(gM = 0.441, gSD = 2.49, Mdn = 0.383, IQR = 0.438, LQ = 0.267, UQ = 0.704)*, however not between small and large. 2-way significance was observed between the trial repetition and block size *F (2, 32) = 9.183 p = 0.001*, with pairwise t-tests showing interactions where the smallest block of first trial was slower than both the large and medium block. All three blocks of the second trial were grasped faster than the first trials smallest block which can be seen in table **#**. Pairwise T-Test effects between haptic and block size were shown, with the lack of haptics producing faster results between small *(gM = 0.422, gSD = 2.168, Mdn = 0.367, IQR = 0.338, LQ = 0.267, UQ = 0.604)* and large *(gM = 0.397, gSD = 2.615, Mdn = 0.395, IQR = 0.375, LQ = 0.262, UQ = 0.637)*, and medium *(gM = 0.408, gSD = 2.125, Mdn = 0.383, IQR = 0.333, LQ = 0.267, UQ = 0.6)* and large blocks.

Interaction	**gMean**	**gSD**	**Median**	**IQR**	**LQ**	**UQ**
Trial 1, Large Block	0.448	2.557	0.433	0.383	0.267	0.65
Trial 1, Medium Block	0.388	2.557	0.383	0.357	0.26	0.617
Trial 1, Small Block	0.465	2.378	0.383	0.471	0.267	0.738
Trial 2, Large Block	0.364	2.422	0.358	0.387	0.233	0.620
Trial 2, Medium Block	0.385	2.097	0.383	0.317	0.25	0.567
Trial 2, Small Block	0.419	2.602	0.383	0.375	0.233	0.608

Table 8 Table showing the differences in block size statistics when grouped by trial.

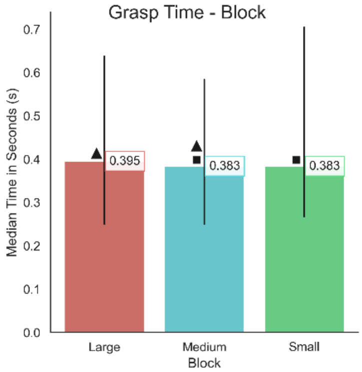


Figure 44 Graph showing the median, lower quantile, and upper quantile of grasp times for the BS task, with pairwise comparison grouped by block size. ▲■ denotes significant pairwise interactions.

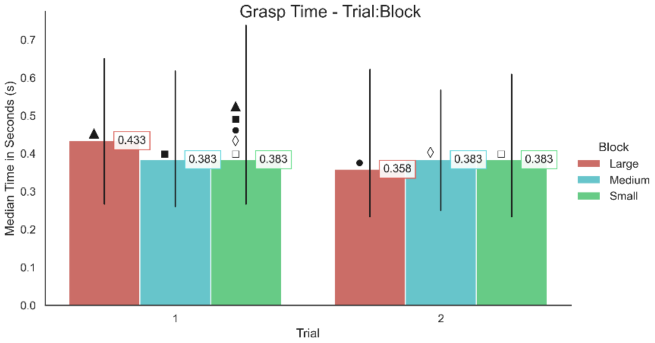


Figure 45 Graph showing the median, lower quantile, and upper quantile of grasp times for the BS task, with pairwise comparison grouped by trial repetition and block size. ▲■●◊□ denotes significant pairwise interactions.

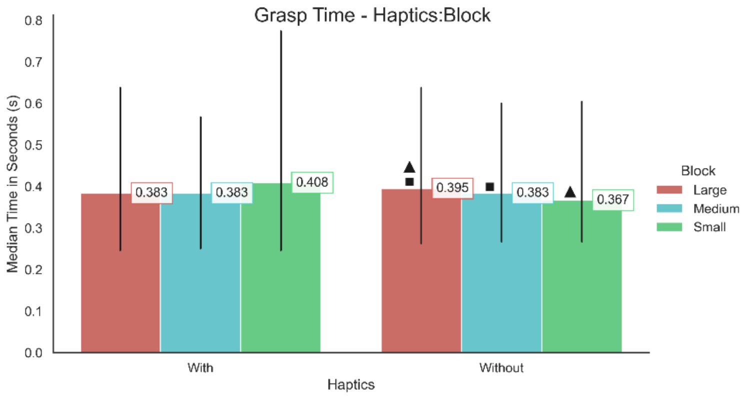


Figure 46 Graph showing the median, lower quantile, and upper quantile of grasp times for the BS task, with pairwise comparison grouped by haptic application and block size. ▲■ *denotes significant pairwise interactions.*

The 3-way combination of haptics, input, and block size resulted in multiple pairwise t-test significance. Specifically, the addition of haptics to hand tracking resulted in slower times for the largest block *(gM = 0.327, gSD = 2.16, Mdn = 0.292, IQR = 0.304, LQ = 0.217, UQ = 0.521)* compared to without *(gM = 0.283, gSD = 2.451, Mdn = 0.308, IQR = 0.192, LQ = 0.217, UQ = 0.408)*, but was not an effect for other block sizes. There was no direct effect of block size or haptic addition to the timings when using controllers. Hand tracking was shown to be significantly faster than controllers when haptics were applied for both the largest *(controllers gM = 0.516, gSD = 2.496, Mdn = 0.55, IQR = 0.521, LQ = 0.317, UQ = 0.838) (hands gM = 0.327, gSD = 2.16, Mdn = 0.292, IQR = 0.304, LQ = 0.217, UQ = 0.521)* and smallest block *(controllers gM = 0.64, gSD = 2.732, Mdn = 0.552, IQR = 0.742, LQ = 0.379, UQ = 1.121) (hands gM = 0.333, gSD = 2.616, Mdn = 0.317, IQR = 0.321, LQ = 0.212, UQ = 0.533)*, however this difference was not reported for the medium block.

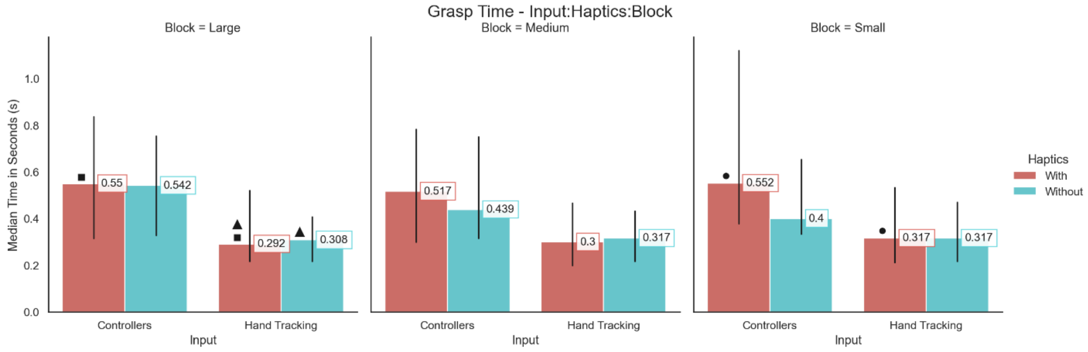


Figure 47 Graph showing the median, lower quantile, and upper quantile of grasp times for the BS task, with pairwise comparison grouped by input type, haptic application, and block size. ▲■● *denotes significant pairwise interactions.*

A 3-way interaction was found between trial, input, and block size. Hand tracking was shown to be faster when grasping the largest block in the second trial *(gM = 0.295, gSD = 2.337, Mdn = 0.308, IQR = 0.258, LQ = 0.212, UQ = 0.471)* compared to the first trial *(gM = 0.315, gSD = 2.288, Mdn = 0.3, IQR = 0.237, LQ = 0.217, UQ = 0.454)*, Hand tracking showed slower times during the second trial when grasping the smallest block *(gM = 0.329, gSD = 2.579, Mdn = 0.308, IQR = 0.367, LQ = 0.167, UQ = 0.533)* compared to the largest, but did not display significance during the first trial. Controllers showed slower times during the first trial for both the large *(gM = 0.639, gSD = 2.496, Mdn = 0.562, IQR = 0.542, LQ = 0.4, UQ = 0.942)* and medium block *(gM = 0.534, gSD = 2.315, Mdn = 0.517, IQR = 0.521, LQ = 0.329, UQ = 0.85)*, when compared to the hand tracking's large block during the second trial.

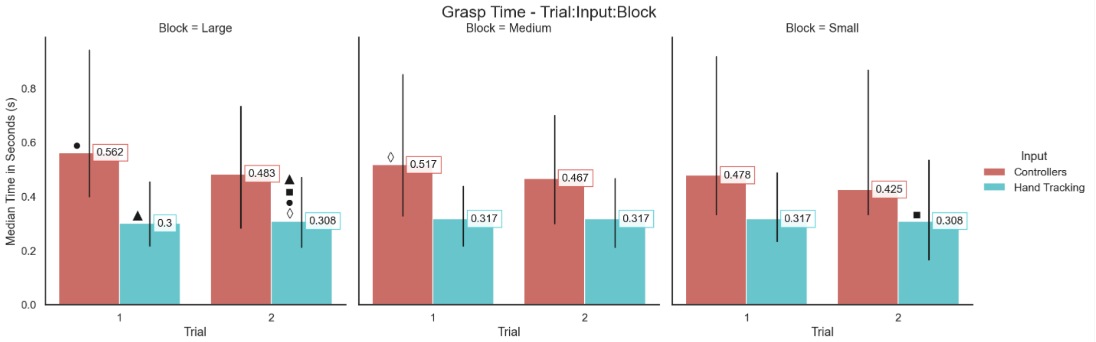


Figure 48 Graph showing the median, lower quantile, and upper quantile of grasp times for the BS task, with pairwise comparison grouped by trial repetition, input type, and block size. ▲■●◊ denotes significant pairwise interactions.

4.4.1.4 Object Release Time

Release time was recorded, in seconds, as the time between an object being released or no longer moving and the current input method leaving a radius of 10cm of the object's extents.

Overall release times were *(gM = 0.525, gSD = 1.802, Mdn = 0.533, IQR = 0.35, LQ = 0.383, UQ = 0.733)*, with a heavy positive skew *(3.722)*.

The effects of haptics, input type, block size, and task repetition were compared. As with object grasp times, the stage main effect was independently analysed. The stage effect was shown to be significant *F (1, 16) = 54.074 p = <0.001* and had a faster release during the dismantling stage *(gM = 0.455, gSD = 1.731, Mdn = 0.467, IQR = 0.292, LQ = 0.336, UQ = 0.627)* than the stacking stage *(gM = 0.606, gSD = 1.811, Mdn = 0.617, IQR = 0.354, LQ = 0.45, UQ = 0.804)*.

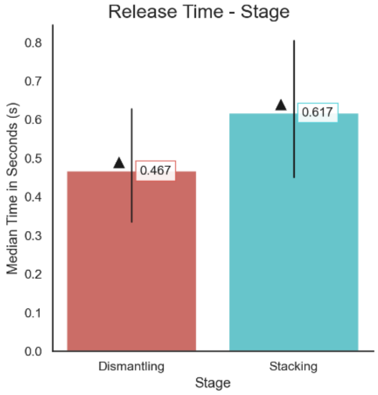


Figure 49 Graph showing the median, lower quantile, and upper quantile of release times for the BS task, with pairwise comparison grouped by stage of task. ▲ denotes significant pairwise interaction.

Main effects of haptics *F (1, 16) = 7.532 p = 0.014*, input *F (1, 16) = 34.607 p = <0.001*, and block *F (2, 32) = 6.123 p = 0.006* were shown to be significant. The addition of haptics slowed the release time *(gM = 0.546, gSD = 1.834, Mdn = 0.55, IQR = 0.383, LQ = 0.4, UQ = 0.783)* compared to without *(gM = 0.505, gSD = 1.767, Mdn = 0.517, IQR = 0.317, LQ = 0.367, UQ = 0.683)*. While controllers were significantly faster at releasing objects *(gM = 0.463, gSD = 1.706, Mdn = 0.467, IQR = 0.271, LQ = 0.362, UQ = 0.633)* compared to hands *(gM = 0.595, gSD = 1.849, Mdn = 0.617, IQR = 0.356, LQ = 0.45, UQ = 0.806)*. Although block size was a main effect, the 3 degrees of freedom required pairwise t-test analysis, which reported no significant differences.

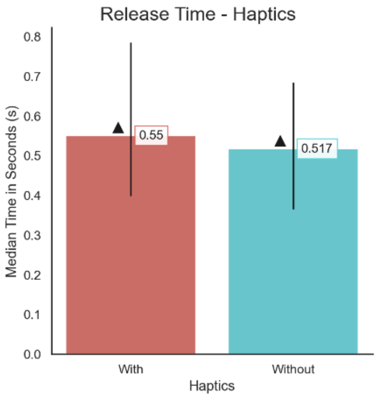


Figure 50 Graph showing the median, lower quantile, and upper quantile of release times for the BS task, with pairwise comparison grouped by haptic application. ▲ denotes significant pairwise interaction.

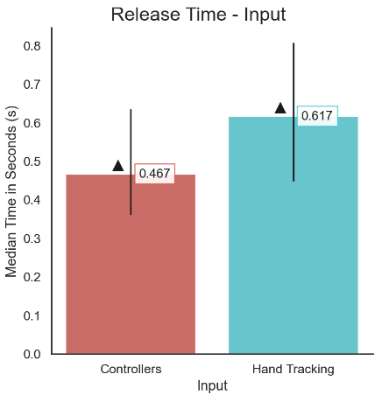


Figure 51 Graph showing the median, lower quantile, and upper quantile of release times for the BS task, with pairwise comparison grouped by input type. ▲ denotes significant pairwise interaction.

A 3-way significant effect between haptics, input type, and block size was reported $F(2, 32) = 4.868\ p = 0.014$, however no corresponding pairwise t-test reported any significant difference. No other ANOVA reported any significance.

4.4.1.5 Object Movement Time

Object movement time was recorded as the time in seconds between the object being grasped and then released. This was primarily recorded to understand differences in overall time taken to perform accurate movements.

Object movement times as a whole were $(gM = 1.417, gSD = 1.596, Mdn = 1.409, IQR = 0.733, LQ = 1.117, UQ = 1.85)$, with an incredibly heavy right skew (6.148).

We compared the effects of haptics, input type, block size, task repetition, and task stage. Block size was the only main effect to show any significance $F(2, 32) = 25.339\ p = < 0.001$, with corresponding pairwise t-tests reporting that the smallest block was slower $(gM = 1.574, gSD = 1.603, Mdn = 1.6, IQR = 0.738, LQ = 1.317, UQ = 2.054)$ than both the medium $(gM = 1.402, gSD = 1.426, Mdn = 1.408, IQR = 0.593, LQ = 1.111, UQ = 1.704)$ and largest block $(gM = 1.289, gSD = 1.711, Mdn = 1.292, IQR = 0.542, LQ = 1.012, UQ = 1.554)$.

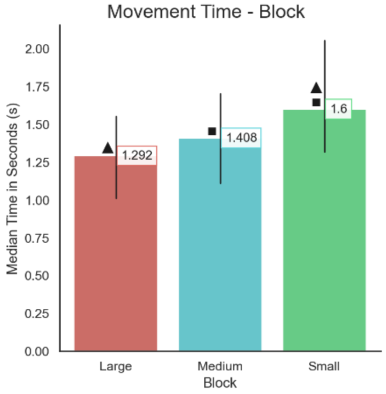


Figure 52 Graph showing the median, lower quantile, and upper quantile of movement times for the BS task, with pairwise comparison grouped by block size. ▲■ denotes significant pairwise interactions.

A 3-way significance was observed between haptics, input type, and block size $F(2, 32) = 3.619\ p = 0.038$, however no follow up pairwise t-test was reported. Significance was also reported between trial, block size, and task stage *F (2, 32) = 3.409 p = 0.045*, but once again no pairwise t-test significance was observed.

4.4.1.6 Block Stacking Accuracy

Stacking accuracy was recorded in two parts, with the first being the difference in centroid distance between the largest block and the medium size block, and the medium block centroid and the smallest size block once placed. The first accuracy value will be referred to as A1 and the second as A2. Accuracy values were analysed both independently and together to understand if there were any significant differences between. Distance values are reported in centimetres.

Accuracy values for A1 were *(gM = 0.896, gSD = 2.186, Mdn = 1.02, IQR = 0.959, LQ = 0.597, UQ = 1.556)*, with a positive skew *(0.494)*. A2 was *(gM = 0.739, gSD = 1.935, Mdn = 0.797, IQR = 0.703, LQ = 0.497, UQ = 1.2)*, with a positive skew *(0.531)*. Finally, between values was *(gM = 0.813, gSD = 2.073, Mdn = 0.907, IQR = 0.845, LQ = 0.537, UQ = 1.383)*, with a positive skew *(0.670)*.

Both accuracy values were compared by the main effects of input type, haptics, and trial repetition. This was expanded to included accuracy type when comparing between. The main effect of input type was shown to be significant, with controllers being more accurate for A1 at placing blocks *(gM = 0.746, gSD = 2.238, Mdn = 0.949, IQR = 0.736, LQ = 0.53, UQ = 1.266)* compared to hands *(gM = 1.076, gSD = 2.051, Mdn = 1.283, IQR = 1.347, LQ = 0.717, UQ = 2.064)*. However no further significant values were found for either A1 or A2.

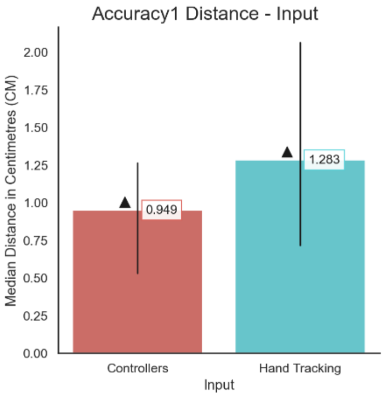


Figure 53 Graph showing the median, lower quantile, and upper quantile of A1 distances for the BS task, with pairwise comparison grouped by input type. ▲ denotes significant pairwise interaction.

When directly comparing accuracies, input was also found to be significant *F (1, 16) = 6.378 p = 0.022* where controllers *(gM = 0.72, gSD = 2.067, Mdn = 0.799, IQR = 0.692, LQ = 0.493, UQ = 1.185)* were once again more accurate than hand tracking *(gM = 0.92, gSD = 2.041, Mdn = 1.013, IQR = 0.96, LQ = 0.616, UQ = 1.576)*. The difference between accuracies was shown to have a significant effect *F (1, 16) = 7.887 p = 0.013*, with A1 *(gM = 0.896, gSD = 2.186, Mdn = 1.02, IQR = 0.959, LQ = 0.597, UQ = 1.556)* being less accurate than A2 *(gM = 0.739, gSD = 1.935, Mdn = 0.797, IQR = 0.703, LQ = 0.497, UQ = 1.2)*. The 2-way effect of trial repetition and input type was shown to have a significant effect *F (1, 16) = 5.272 p = 0.036*, however no pairwise t-tests reported any significant interactions.

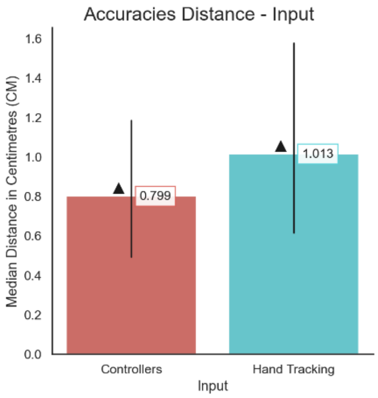


Figure 54 Graph showing the median, lower quantile, and upper quantile of between accuracy type values for the BS task, with pairwise comparison grouped by input type. ▲ denotes significant pairwise interaction.

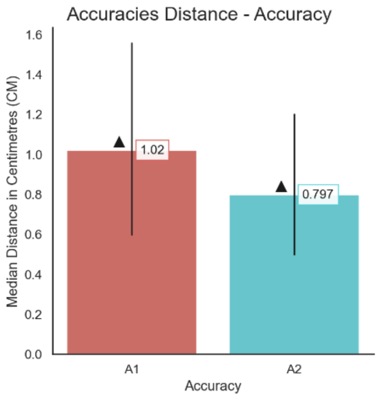


Figure 55 Graph showing the median, lower quantile, and upper quantile of between accuracy type values for the BS task, with pairwise comparison grouped by accuracy type. ▲ denotes significant pairwise interaction.

4.4.1.7 Object Travel Distance

The object travel distance was recorded as the cumulative distance in which a block had moved during the entire time it had be grasped, up until the point of release. This was calculated by measuring the distance between the positions of the block between each frame, and then combining each value to reach a total. Values are reported in centimetres.

Object travel distances were *(gM = 38.694, gSD = 1.379, Mdn = 38.393, IQR = 9.383, LQ = 34.244, UQ = 43.627)*, with a heavy positive skew *(4.775)*.

We compared the main effects of haptics, input type, trial repetition, and block size. Task stage was independently analysed and was reported to be significant *F (1, 16) = 10.919 p = 0.004*, with the stacking segment resulting in shorter distances *(gM = 37.031, gSD = 1.492, Mdn = 36.905, IQR = 8.675, LQ = 33.141, UQ = 41.815)* than the dismantling stage *(gM = 40.432, gSD = 1.229, Mdn = 40.564, IQR = 9.195, LQ = 35.802, UQ = 44.997)*.

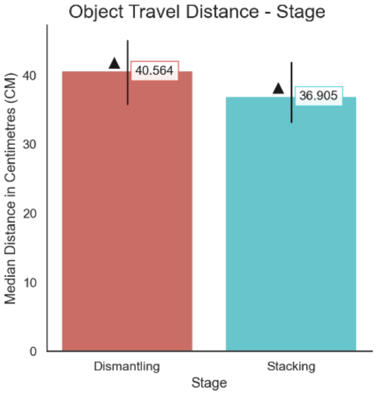


Figure 56 Graph showing the median, lower quantile, and upper quantile of object travel distances for the BS task, with pairwise comparison grouped by stage. ▲ denotes significant pairwise interaction.

The trial main effect reported significance *F (1, 16) = 11.34 p = 0.004*, with shorter distances for the second trial *(gM = 37.918, gSD = 1.268, Mdn = 37.92, IQR = 8.617, LQ = 33.795, UQ = 42.411)* compared to the first *(gM = 39.486, gSD = 1.471, Mdn = 39.099, IQR = 11.041, LQ = 34.66, UQ = 45.701)*. No 4-way or 3-way interactions were reported. A 2-way interaction was reported between trial repetition and block size *F (2, 32) = 3.373 p =*

0.047. Follow-up pairwise t-tests reported a single interaction where the large block *(gM = 38.291, gSD = 1.259, Mdn = 37.394, IQR = 9.348, LQ = 33.52, UQ = 42.868)* reported larger values than the medium block *(gM = 37.278, gSD = 1.338, Mdn = 38.155, IQR = 7.667, LQ = 34.108, UQ = 41.775)* in the second trial. No other factors were reported as significant, nor were any follow up t-tests.

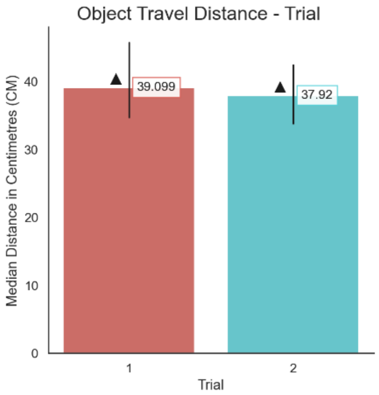


Figure 57 Graph showing the median, lower quantile, and upper quantile of object travel distances for the BS task, with pairwise comparison grouped by trial repetition. ▲ denotes significant pairwise interaction.

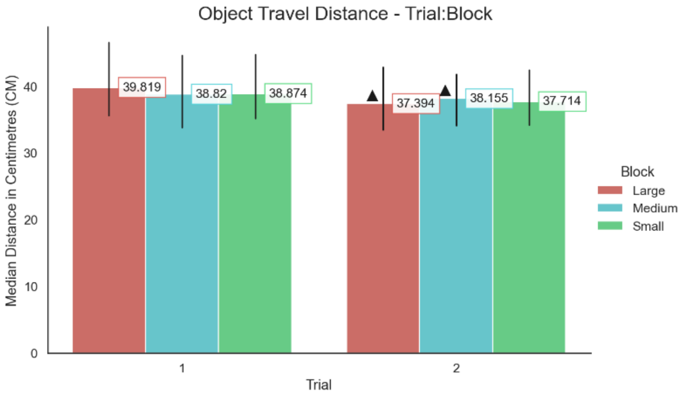


Figure 58 Graph showing the median, lower quantile, and upper quantile of object travel distances for the BS task, with pairwise comparison grouped by trial and block size. ▲ denotes significant pairwise interaction.

4.4.1.8 Object Rotation

Object rotation was recorded as the total cumulative degrees in which a block had been rotated during the entire time it had be grasped, until the point of release. This was calculated by measuring the difference in rotation degrees between frames of the currently grasped object, and then combined to report the final value.

Object rotation values as a whole were *(gM = 90.627, gSD = 1.642, Mdn = 87.301, IQR = 50.644, LQ = 66.578, UQ = 117.222)*, with an incredibly heavy skew *(10.427)*.

We analysed the effect of haptics, input type, trial repetition, and block size. As with other statistics, the stage variable was independently analysed where it was shown to be significant *F (1, 16) = 17.123 p = 0.001*, with the stacking stage having larger rotational values *(gM = 97.304, gSD = 1.751, Mdn = 93.812, IQR = 56.007, LQ = 72.652, UQ = 128.658)* than the dismantling stage *(gM = 84.409, gSD = 1.508, Mdn = 79.189, IQR = 44.306, LQ = 62.579, UQ = 106.884)*.

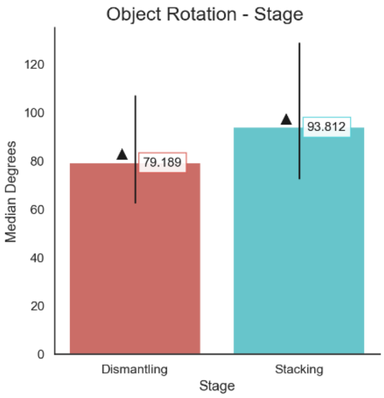
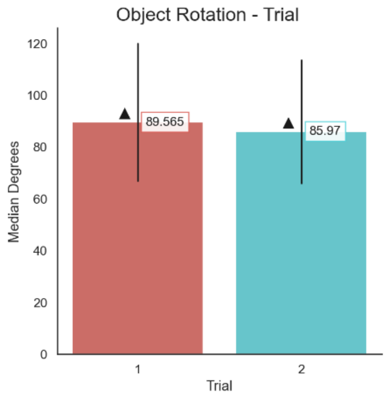


Figure 59 Graph showing the median, lower quantile, and upper quantile of object rotation values for the BS task, with pairwise comparison grouped by stage. ▲ denotes significant pairwise interaction.

Significance was found for the main effects of trial *F (1, 16) = 6.11 p = 0.025*, input *F (1, 16) = 16.554 p = 0.001*, and block size *F (2, 32) = 4.28 p = 0.023*. The second trial had smaller rotations *(gM = 87.906, gSD = 1.535, Mdn = 85.97, IQR = 47.514, LQ = 65.891, UQ = 113.406)* compared to the first *(gM = 93.433, gSD = 1.74, Mdn = 89.565, IQR = 52.93, LQ = 66.962, UQ = 119.892)*. Controllers had smaller rotation values *(gM = 78.918, gSD = 1.65, Mdn = 72.604, IQR = 38.588, LQ = 59.7, UQ = 98.287)* compared to hand tracking *(gM = 104.074, gSD = 1.569, Mdn = 103.593, IQR = 49.009, LQ = 80.339, UQ = 129.348)*. Pairwise t-tests showed no significant differences between block sizes, required due to the increased degrees of freedom.



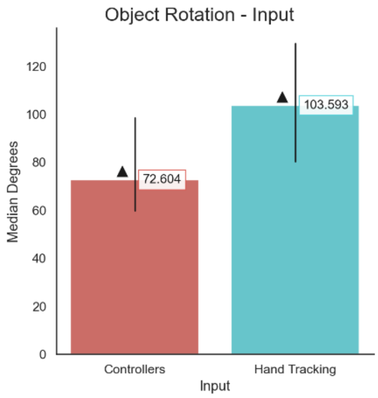


Figure 60 Graph showing the median, lower quantile, and upper quantile of object rotation values for the BS task, with pairwise comparison grouped by trial repetition. ▲ denotes significant pairwise interaction.

Figure 61 Graph showing the median, lower quantile, and upper quantile of object rotation values for the BS task, with pairwise comparison grouped by input type. ▲ denotes significant pairwise interaction.

3-way significance was found between haptics, input, and block size *F (2, 32) = 7.564 p = 0.002*, where pairwise t-tests reported that the controllers with haptics for the medium block produced smaller rotations *(gM = 79.195, gSD = 1.424, Mdn = 72.086, IQR = 28.882, LQ = 64.47, UQ = 93.353)*, compared to the large block *(gM = 77.264, gSD = 1.737, Mdn = 67.14, IQR = 35.49, LQ = 55.199, UQ = 90.688)*. No further significant pairwise interactions were observed. A 3-way significance was also found between trial repetition, haptics, and input type *F (1, 16) = 5.045 p = 0.039*. Pairwise t-tests reported a difference between direct counter parts of hand tracking in the first trial without haptics *(gM = 101.067, gSD = 1.693, Mdn = 100.524, IQR = 41.956, LQ = 77.696, UQ = 119.652)* having larger rotational values than the controller input in the second trial with haptics *(gM = 73.629, gSD = 1.493, Mdn = 68.673,*

$IQR = 28.745, LQ = 57.261, UQ = 86.006$). No further t-test or 3-way significance was reported.

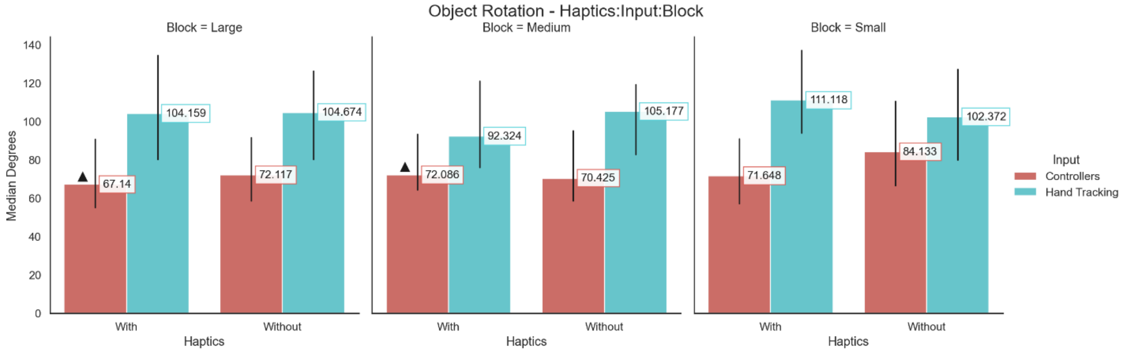


Figure 62 Graph showing the median, lower quantile, and upper quantile of object rotation values for the BS task, with pairwise comparison grouped by haptic application, input type, and block size. ▲ denotes significant pairwise interaction.

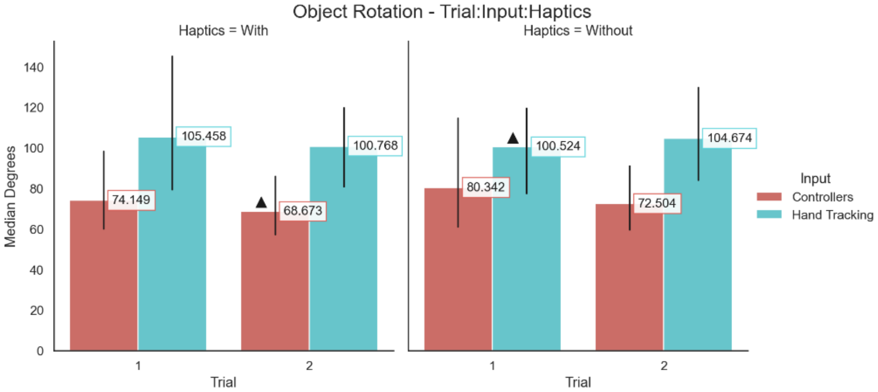


Figure 63 Graph showing the median, lower quantile, and upper quantile of object rotation values for the BS task, with pairwise comparison grouped by trial repetition, input type, and haptic application. ▲ denotes significant pairwise interaction.

A 2-way significance was reported between trial and block size $F\ (2,\ 32) = 4.042\ p = 0.027$, where t-tests reported larger rotations for all three block sizes in the first trial when compared to the medium block in the second trial. Differences were also reported between the second trials largest block and the medium, having smaller values, and small block, having larger values. The values between blocks can be viewed in table #.

Test	**Mean**	**Median**	**IQR**	**LQ**	**UQ**
Trial 1, Large Block	126.207	88.366	58.315	62.463	120.779
Trial 1, Medium Block	98.671	81.469	50.882	67.393	118.275
Trial 1, Small Block	107.942	98.624	54.96	70.618	125.578
Trial 2, Large Block	96.449	80.146	42.692	64.17	106.862
Trial 2, Medium Block	93.44	86.013	41.19	66.793	107.983
Trial 2, Small Block	101.402	90.611	49.571	68.01	117.58

Table 9 Table showing object rotational values for the different block sizes grouped by trial.

4.4.1.9 Object Grasp Pose

The object grasp pose was recorded as the positioning of the hand when it was initially grasping the object. This was categorised into one of three different poses, either a two finger pinch (1), a multi-finger pinch of at least the thumb and two other fingers (2), or a whole hand grasp (3). This was recorded to understand differences between overall object size and the reactive nature of objects.

GEE analysis was performed to understand the effect of trial repetition, haptic feedback addition, input type, task type, and block size on the different hand poses.

No statistically significant 5-way interaction was reported $(p > 0.05)$. A single significant 4-way interaction was found between trial repetition, haptic application, input type, and block size, $Wald\ \chi2(2) = 6.794,\ p = 0.033$. No statistically significant 3-way interactions were reported. Statistically significant 2-way interactions were reported for input type and block size, $Wald\ \chi2(2) = 12.269,\ p = 0.002$, input type and task stage $Wald\ \chi2(1) = 4.902,\ p = 0.027$, and trial repetition and input type, $Wald\ \chi2(1) = 7.587,\ p = 0.006$. Statistical significance was observed for the main effects of task stage, $Wald\ \chi2(1) = 17.470,\ p < 0.001$, and block size $Wald\ \chi2(2) = 30.827,\ p < 0.001$.

Parameter estimation reported multiple significant interactions. These are reported over a scale where a lower value designates being closer to a two finger pinch (closer to zero), while a high value denotes being closer to a whole hand grasp (significantly higher than one). The 4-way significant odds of controllers during the first trials' stacking stage, when interacting with the medium block using a whole hand grasp instead of a two finger pinch was $0.256\ (95\%\ CI,\ 0.071 - 0.929)$ times that of the smallest block, $Wald\ \chi2(1) = 4.293,\ p = 0.038$. No other 4-way or 3-way significant estimations were reported. Controllers showed significant odds of interacting with the largest block with a whole hand grasp over a two finger pinch of $0.211\ (95\%\ CI,\ 0.049 - 0.911)$ times that of the smallest block $Wald\ \chi2(1) = 4.345,\ p = 0.037$. The likelihood of the largest block being grasped by the whole hand over the pinch was $5.559\ (95\%\ CI,\ 1.757\text{-}17.593)$ times that of the smallest block, $Wald\ \chi2(1) = 8.517,\ p = 0.004$. This was also the case for medium block sizes, where they were likely to be grasped using the whole hand over the pinch $2.289\ (95\%\ CI,\ 1.051 - 4.983)$ times that of the smallest block, $Wald\ \chi2(1) = 4.353,\ p = 0.037$. The stacking stage of the task with hand poses more likely to be a whole hand grasp than pinch was $2.846\ (95\%\ CI,\ 1.163 - 6.962)$ times that of the dismantling stage, $Wald\ \chi2(1) = 5.251,\ p = 0.022$. Controllers had hand pose values $3.746\ (95\%\ CI,\ 1.056 - 13.287)$ times higher than hand tracking, $Wald\ \chi2(1) = 3.746,\ p = 0.041$. No other significant values were reported.

4.4.1.10 Failure Occurrences

Throughout the study, the number of failures were recorded that occurred whilst users conducted the block stacking task under the varying conditions. Events such as the inability to grasp a block in a single attempt, dropping blocks during placement, and the accidental toppling of the stack were all considered as failures.

GEE analysis was performed to determine the effect of trial repetition, inclusion of haptic feedback, input type, task type, and block size on the frequency of failures. The QIC and QICC goodness of fit values for the GEE model were 986.025, and 957.912 respectively. The overall values were $(M = 1.010,\ SD = 2.217)$.

No statistically significant 5-way interaction was found $(p > 0.05)$. Statistically significant 4-way interactions were found between trial repetition, haptic application, task stage, and block size, $Wald\ \chi\ 2(2) = 14.459,\ p < 0.001$, and trial repetition, input type, task stage, and block size, $Wald\ \chi\ 2(2) = 7.07,\ p = 0.029$. Numerous statistically significant 3-way interactions were found, as summarised in Table 10. Statistically significant 2-way interactions were found to exist between trial repetition and input type, $Wald\ \chi\ 2(1) = 4.772,\ p = 0.029$, and between input type and block size $Wald\ \chi\ 2(2) = 6.23,\ p = 0.044$. The inclusion of haptic feedback had a significant effect on the number of failures produced, $Wald\ \chi\ 2(1) = 6.764,\ p= 0.009$, as did input type, $Wald\ \chi\ 2(1) = 5.125,\ p = 0.024$, and block size, $Wald\ \chi\ 2(2) = 32.523,\ p < 0.001$.

Test	Wald Chi-Square	DoF	Sig
Trial:Haptics:Stage:Block	14.459	2	<0.001
Trial:Input:Stage:Block	7.070	2	0.029
Input:Stage:Block	15.050	2	0.001
Haptics:Input:Block	6.117	2	0.047
Haptics:Input:Stage	6.467	1	0.011
Trial:Haptics:Stage	4.658	1	0.031
Input:Block	6.230	2	0.044
Trial:Input	4.772	1	0.029
Block	32.523	2	<0.001
Input	5.125	1	0.024
Haptics	6.764	1	0.009

Table 10 Table showing the statistically significant interactions between the different independent variables and effects of individual variable on overall failure rates during BS.

Pairwise comparisons were conducted showing that conditions where haptics was not present produced fewer failures *(M = 0.62, 95% CI, 0.51 - 0.77)* than conditions with haptic feedback *(M = 0.92, 95% CI, 0.68 - 1.23)*, *Wald χ 2(1) = 5.221, p = 0.022*. The use of hand tracking for input type significantly reduced the number of mean failures *(M = 0.65, 95% CI, 0.7 - 1.12)* compared to controller-based input *(M = 0.89, 95% CI, 0.5 - 0.84)*, *Wald χ 2(2) = 5.019, p = 0.025*. Pairwise comparisons between block size highlighted that interactions with the largest block produced significantly fewer failures *(M = 0.39, 95% CI, 0.28 - 0.55)*, than the medium sized block *(M = 0.94, 95% CI, 0.75 -1.19)*, and the smallest block *(M = 1.17, 95% CI, 0.89 - 1.56)*, *Wald χ2(2) = 32.480, p < 0.001*. Failure rates were not significantly different between medium and small block interactions.

4.4.1.11 Task Load Index Ratings

Task load index ratings were recorded after each task, with the participant choosing their ratings from a 7-point Likert scale (0 being best, 6 worst). These ratings were modelled after the NASA TLX ratings, which questioned the participant on their opinions of the tasks required mental capacity, the amount of physical strain, how the task affected their pace and timing, how successful their performance was, the overall amount of required effort needed, and finally their personal level of frustration with the task.

We chose to analyse the ratings in two different methods. First by having the overall TLX values being converted into a single mean value per data point, thus reducing main effects. Then secondly by comparing the individual TLX categories as a main effect. This resulted in tests between input type, haptic condition, trial repetition, and in the second case, TLX factors. We will cover the mean ratings first, followed by the individual TLX factors.

Overall TLX values reported *(gM = 1.264, gSD = 2.376, Mdn = 1.5, IQR = 1.708, LQ = 0.667, UQ = 2.375)*, with a positive skew *(0.341)* for means and a heavier positive skew *(0.640)* for individual factors.

Mean TLX values reported significant 2-way effects between trial and input type *F (1, 16) = 6.231 p = 0.024*, however resulting pairwise t-tests did not report any significant differences. No other 3-way or 2-way significance was reported. The haptic main effect reported that the addition of haptics increased the overall workload *(without haptics gM = 1.112, gSD = 2.395, Mdn = 1.167, IQR = 1.375, LQ = 0.625, UQ = 2.0) (with haptics gM = 1.439, gSD = 2.327, Mdn = 1.833, IQR = 1.583, LQ = 0.958, UQ = 2.542)*. No other significant main effect was reported.

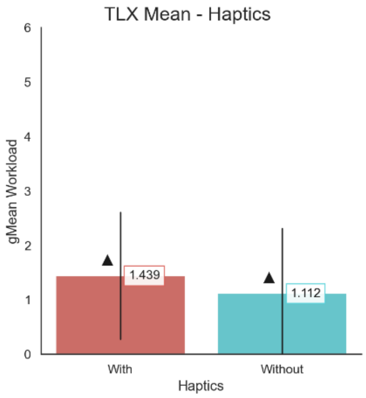


Figure 64 Graph showing the geometric mean and geometric standard deviation of mean TLX values for the BS task, with pairwise comparison grouped by haptic application. ▲ *denotes significant pairwise interaction.*

When comparing individual factors, a 4-way interaction between all the effects was reported *F (5, 80) = 3.126 p = 0.013*, however no follow-up pairwise t-test was reported as significant. A single 2-way interaction was once again reported between trial and input, however when analysing this set of data there were multiple pairwise t-test interactions, displayed in Figure 65 and values reported in Table 11.

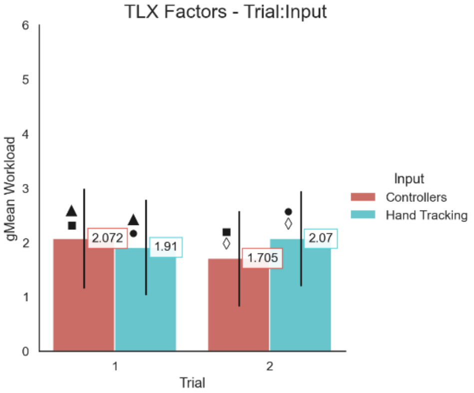


Figure 65 Graph showing the geometric mean and geometric standard deviation of mean TLX values for the BS task, with pairwise comparison grouped by haptic application. ▲■●◊ *denotes significant pairwise interaction.*

Trial	Input	gMean	gSD	Median	IQR	LQ	UQ
1	**Hand Tracking**	1.910	1.731	1	2	0	2
	Controllers	2.072	1.806	2	2	1	3
2	**Hand Tracking**	2.070	1.724	1	3	0	3
	Controllers	1.705	1.734	1	2	0	2

Table 11 Table showing the overall TLX values from the BS task when grouped by trial and input type, and when individually analysing the TLX factors instead of by mean.

The main effect of haptics was still highlighted as being significant, reporting *F (1, 16) = 5.927 p = 0.027*, with haptics showing a higher overall workload *(gM = 2.061, gSD = 1.77, Mdn = 1.0, IQR = 3.0, LQ = 0.0, UQ = 3.0)* than without *(gM = 1.819, gSD = 1.735, Mdn = 1.0, IQR = 2.0, LQ = 0.0, UQ = 2.0)*. The TLX rating main effect was reported as significant *F (5, 80) = 10.338 p = < 0.001*. Following pairwise t-tests reported interactions between the effort factor and all bar the physical factor, between frustration and temporal ratings, between mental load and physical, and performance, and finally between temporal and mental, physical, and performance. Values of which can be found in Table 12, with interactions show in Figure 66.

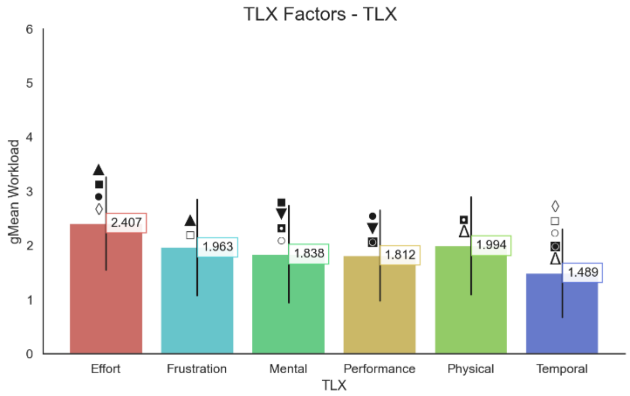


Figure 66 Graph showing the geometric mean and geometric standard deviation of mean TLX values for the BS task, with pairwise comparison grouped by the different TLX question categories. ▲■●◊▼◘○◙Δ denotes significant pairwise interactions.

TLX Question	gMean	gSD	Median	IQR	LQ	UQ
Effort	2.407	1.715	2	3	1	4
Frustration	1.963	1.773	1	3	0	3
Mental	1.838	1.796	1	2	0	2
Performance	1.812	1.668	2	1	1	2
Physical	1.994	1.798	2	2	1	3
Temporal	1.489	1.625	0	1	0	1

Table 12 Table showing the different TLX question statistics, when analysing individual factors, during the BS task.

4.4.2 Object Sorting Task

4.4.2.1 Overall Task Completion Time

The time, in seconds, to complete the entire task was recorded. Timing commenced when the participant pressed the continue button to start the task, and ended when the final block was sorted into the correct container.

Task completion times were *(gM = 48.133, gSD = 1.425, Mdn = 45.95, IQR = 21.642, LQ = 37.567, UQ = 59.208)*, with a heavy positive skew *(1.388)*.

We conducted analysis to understand the effect of haptics, input type, and task repetition. No 3-way or 2-way significance was reported for the task completion time. The main effect of trial was reported as significant *F (1, 16) = 20.994 p = < 0.001*, where the second trial was completed faster *(gM = 43.303, gSD = 1.356, Mdn = 42.658, IQR = 17.187, LQ = 35.246, UQ = 52.433)* than the first *(gM = 53.502, gSD = 1.449, Mdn = 49.842, IQR = 32.45, LQ = 39.542, UQ = 71.992)*. Input main effect was shown to be significant *F (1, 16) = 7.953 p = 0.012* with controllers completing the task faster *(gM = 43.595, gSD = 1.369, Mdn = 42.792, IQR = 17.175, LQ = 34.421, UQ = 51.596)* than hand tracking *(gM = 53.143, gSD = 1.443, Mdn = 50.45, IQR = 27.738, LQ = 40.992, UQ = 68.729)*. No other ANOVA reported significance.

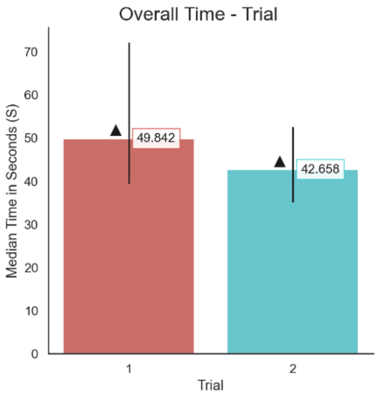


Figure 67 Graph showing the median, lower quantile, and upper quantile of overall task completion time for the OS task, with pairwise comparison grouped by trial repetition. ▲ denotes significant pairwise interaction.

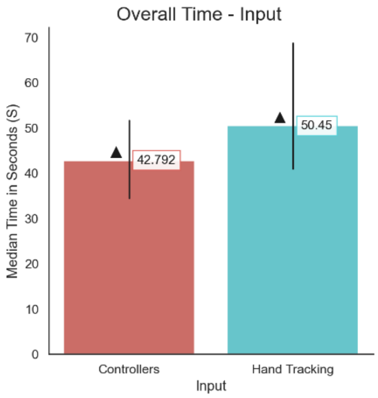


Figure 68 Graph showing the median, lower quantile, and upper quantile of overall task completion time for the OS task, with pairwise comparison grouped by input type. ▲ denotes significant pairwise interaction.

4.4.2.2 Object Grasp Time

Grasp time was recorded, in seconds, as the time between the participants' input method entering a radius of 10cm of the extent of the object and the point of successfully grasping the object. In the case of this task, that meant a constant radius of 12.5cm as the object was always 5cm in width, the same size as the medium block of the BS task. Each object was floating in mid-air at a randomised rotation. This was primarily recorded to understand the differences that the input type have on the speed of interaction with objects.

Overall grasp times were *(gM = 0.417, gSD = 2.579, Mdn = 0.433, IQR = 0.467, LQ = 0.267, UQ = 0.733)*, with an incredibly heavy positive skew *(8.692)*.

Comparisons were made between the effects of haptics, input type, and task repetition. A 3-way interaction was reported between the main effects *F (1, 16) = 7.42 p = 0.015*, with follow-up pairwise t-tests reported multiple significant groups. Firstly, the controllers with haptics during the first trial were found to be the slowest of all groupings *(gM = 0.681, gSD = 2.444, Mdn = 0.667, IQR = 0.721, LQ = 0.421, UQ = 1.142)*, with pairwise t-tests showing significance between it and hand tracking in the second trial with *(gM = 0.375, gSD = 2.073, Mdn = 0.358, IQR = 0.4, LQ = 0.25, UQ = 0.65)* and without haptics *(gM = 0.273, gSD = 2.816, Mdn = 0.317, IQR = 0.312, LQ = 0.204, UQ = 0.517)*. Secondly, that the controllers without haptics within the first trial were shown to be the 2nd slowest grouping *(gM = 0.506, gSD = 2.678, Mdn = 0.5, IQR = 0.496, LQ = 0.333, UQ = 0.829)*, with multiple pairwise t-test significant interactions. They were significantly slower than the hand tracking in the first trial with *(gM = 0.335, gSD = 2.885, Mdn = 0.333, IQR = 0.446, LQ = 0.217, UQ = 0.662)* and without haptics *(gM = 0.337, gSD = 2.351, Mdn = 0.35, IQR = 0.283, LQ = 0.267, UQ = 0.55)*, the same significance being present during the second trial. This was also applicable to the controllers in the second trial without haptics *(gM = 0.497, gSD = 2.241, Mdn = 0.45, IQR = 0.554, LQ = 0.321, UQ = 0.875)*. Hand tracking in the first trial without haptics *(gM = 0.337, gSD = 2.351, Mdn = 0.35, IQR = 0.283, LQ = 0.267, UQ = 0.55)* found pairwise significance between the controllers with haptics in the first *(gM = 0.681, gSD = 2.444, Mdn = 0.667, IQR = 0.721, LQ = 0.421, UQ = 1.142)* and second trial *(gM = 0.463, gSD = 2.385, Mdn = 0.475, IQR = 0.508, LQ = 0.304, UQ = 0.812)*, being faster than both. No further 3-way or 2-way interactions were reported as significant.

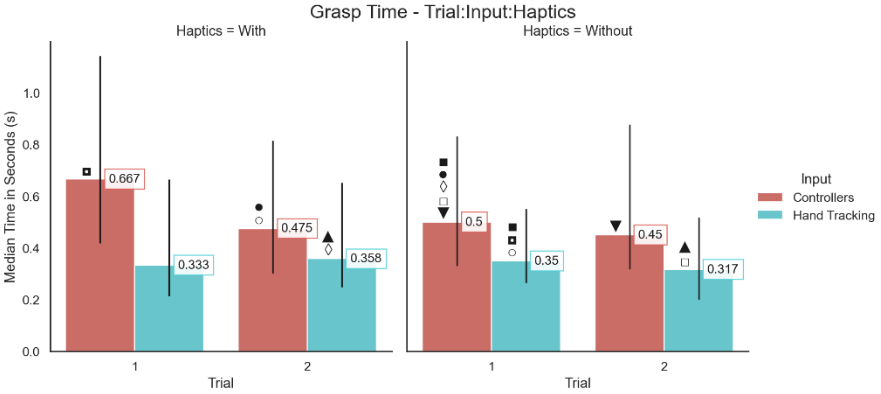


Figure 69 Graph showing the median, lower quantile, and upper quantile of grasp times for the OS task, with pairwise comparison grouped by trial repetition, input type, and haptic application. ▲■●◊▼◘○◙ denotes significant pairwise interactions.

The main effect of input was significant *F (1, 16) = 29.111 p = < 0.001*, with hand tracking being faster *(gM = 0.328, gSD = 2.543, Mdn = 0.333, IQR = 0.371, LQ = 0.229, UQ = 0.6)* than controllers *(gM = 0.531, gSD = 2.459, Mdn = 0.517, IQR = 0.6, LQ = 0.333, UQ = 0.933)*. Haptics were also found to be significant *F (1, 16) = 7.431 p = 0.015*, with the addition of haptics making the grasp time slower *(gM = 0.446, gSD = 2.541, Mdn = 0.467, IQR = 0.55, LQ = 0.267, UQ = 0.817)* compared to without *(gM = 0.39, gSD = 2.608, Mdn = 0.4, IQR = 0.417, LQ = 0.267, UQ = 0.683)*.

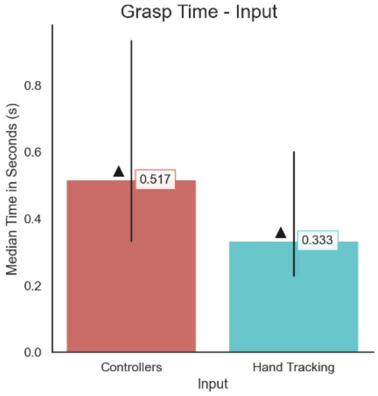


Figure 70 Graph showing the median, lower quantile, and upper quantile of grasp times for the OS task, with pairwise comparison grouped by input type. ▲ denotes significant pairwise interaction.

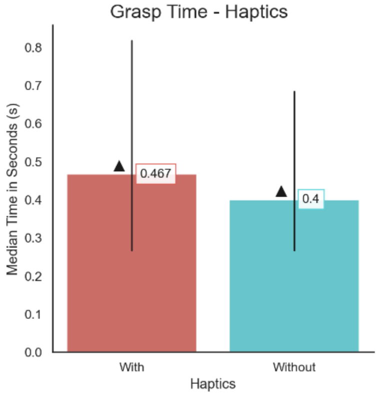


Figure 71 Graph showing the median, lower quantile, and upper quantile of grasp times for the OS task, with pairwise comparison grouped by haptic application. ▲ denotes significant pairwise interaction.

4.4.2.3 Object Release Time

Release time was recorded, in seconds, as the time between an object being released or no longer moving and the current input method leaving a radius of 10cm of the object's extents. This was recorded from a constant radius of 12.5cm as the object was 5cm in width, the same as the medium sized block of the BS task.

Object release times were *(gM = 0.334, gSD = 2.043, Mdn = 0.35, IQR = 0.333, LQ = 0.217, UQ = 0.55)*, with a heavy positive skew *(2.404)*.

Main effects of haptics, input type, and trial repetition were compared to find significance. No 3-way or 2-way significance was found, however each main effect was shown to be individually significant. When haptics *F (1, 16) = 32.771 p = < 0.001* were applied they resulted in faster release times *(gM = 0.259, gSD = 2.108, Mdn = 0.25, IQR =*

0.283, LQ = 0.15, UQ = 0.433) compared to without *(gM = 0.429, gSD = 1.789, Mdn = 0.433, IQR = 0.317, LQ = 0.3, UQ = 0.617)*. The input type showed significance *F (1, 16) = 42.473 p = < 0.001* where the use of controllers were faster *(gM = 0.245, gSD = 1.818, Mdn = 0.25, IQR = 0.217, LQ = 0.167, UQ = 0.383)* to release than hand tracking *(gM = 0.454, gSD = 1.99, Mdn = 0.5, IQR = 0.417, LQ = 0.3, UQ = 0.717)*. The trial repetitions were significant *F (1, 16) = 5.569 p = 0.031*, where the second trial resulted in faster release times *(gM = 0.323, gSD = 2.082, Mdn = 0.35, IQR = 0.317, LQ = 0.217, UQ = 0.533)* than the first *(gM = 0.345, gSD = 2.002, Mdn = 0.367, IQR = 0.35, LQ = 0.217, UQ = 0.567)*.

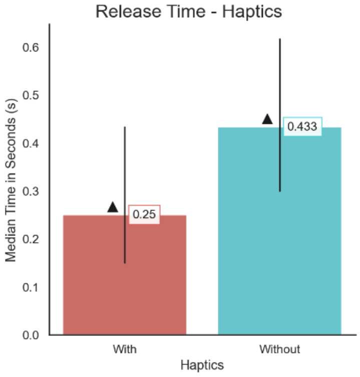


Figure 72 Graph showing the median, lower quantile, and upper quantile of release times for the OS task, with pairwise comparison grouped by haptic application. ▲ denotes significant pairwise interaction.

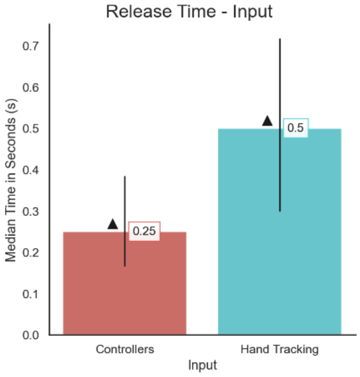


Figure 73 Graph showing the median, lower quantile, and upper quantile of release times for the OS task, with pairwise comparison grouped by input type. ▲ denotes significant pairwise interaction.

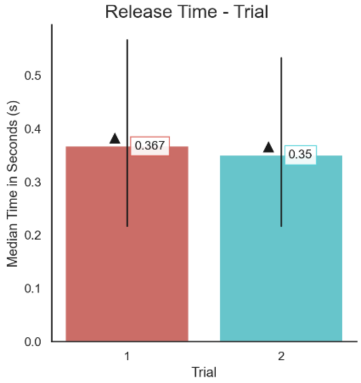


Figure 74 Graph showing the median, lower quantile, and upper quantile of release times for the OS task, with pairwise comparison grouped by trial repetition. ▲ denotes significant pairwise interaction.

4.4.2.4 Object Movement Time

Object movement time was recorded as the time in seconds between the object being grasped and then released. This was primarily recorded to understand differences in overall time taken to perform accurate movements.

Movement times were *(gM = 2.521, gSD = 1.743, Mdn = 2.642, IQR = 1.65, LQ = 1.9, UQ = 3.55)*, with a heavy positive skew *(2.293)*.

Main effects of haptics, input type, and trial repetition were compared. No 3-way significance was observed. A single 2-way significance was found between haptics and trial repetition *F (1, 16) = 5.113 p = 0.038*, however no corresponding pairwise t-test reported any significant. None of the main effects were found to be individually significant.

4.4.2.5 Object Travel Distance

The object travel distance was recorded as the cumulative distance in which a block had moved during the entire time it had be grasped, up until the point of release. This was calculated by measuring the distance between the positions of the block between each frame, and then combining each value to reach a total.

Travel distance of objects was *(gM = 0.647, gSD = 1.721, Mdn = 0.724, IQR = 0.581, LQ = 0.387, UQ = 0.968)*, with a positive skew *(1.086)*.

Each of the main effects of haptics, input type, and trial repetition were contrasted to find any significant differences. No 3-way or 2-way significance was observed for any of the variables. The main effect of input type showed significance *F (1, 16) = 9.566 p = 0.007*, where the hand tracking produced shorter values *(gM = 0.596, gSD = 1.698, Mdn = 0.56, IQR = 0.534, LQ = 0.374, UQ = 0.908)* compared to the controllers *(gM = 0.702, gSD = 1.724, Mdn = 0.806, IQR = 0.59, LQ = 0.431, UQ = 1.021)*.

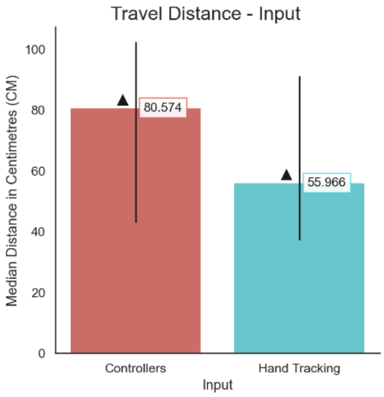


Figure 75 Graph showing the median, lower quantile, and upper quantile of travel distance values for the OS task, with pairwise comparison grouped by input type. ▲ denotes significant pairwise interaction.

4.4.2.6 Object Rotation

Object rotation was recorded as the total cumulative degrees in which a block had been rotated during the entire time it had be grasped, until the point of release. This was calculated by measuring the difference in rotation degrees between frames of the currently grasped object, and then combined to report the final value in degrees.

The overall object rotation values were *(gM = 357.098, gSD = 2.076, Mdn = 414.566, IQR = 405.763, LQ = 202.547, UQ = 608.31)*, with a positive skew *(1.128)*.

We compared the main effects of haptics, input type, and trial repetition between each other to observe any significant differences. No 3-way or 2-way significance was observed. The main effect of input was significantly different *F (1, 16) = 6.735 p = 0.02*, with hand tracking resulting in smaller values *(gM = 336.609, gSD = 2.043, Mdn = 321.082, IQR = 387.295, LQ = 191.045, UQ = 578.34)* compared to controllers *(gM = 378.833, gSD = 2.101, Mdn = 450.321, IQR = 395.909, LQ = 237.936, UQ = 633.845)*. No other main effect reported any significance.

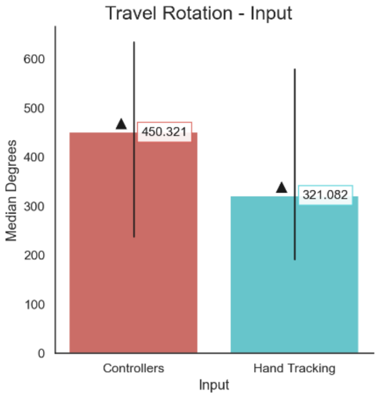


Figure 76 Graph showing the median, lower quantile, and upper quantile of travel rotation degrees for the OS task, with pairwise comparison grouped by input type. ▲ denotes significant pairwise interaction.

4.4.2.7 Object Grasp Pose

The object grasp pose was recorded as the positioning of the hand when it was initially grasping the object. This was categorised into one of three different poses, either a two finger pinch (1), a multi-finger pinch of at least the thumb and two other fingers (2), or a whole hand grasp (3). This was recorded to understand differences between overall object size and the reactive nature of objects.

GEE analysis was performed to understand the effect of trial repetition, haptic feedback addition and input type on the different hand poses.

No statistically significant 3-way or 2-way interactions were reported. The main effect of haptics reported statistical significance $Wald\ \chi2(1) = 7.684,\ p = 0.006$, however, no further main effects reported significance.

No statistically significant parameter estimates were reported.

4.4.2.8 Failure Occurrences

Throughout the study, the number of failures were recorded that occurred whilst users conducted the block stacking task under the varying conditions. Events such as the inability to grasp a block in a single attempt and dropping blocks during sorting were both considered as failures.

GEE analysis was performed to determine the effect of repeated trial, inclusion of haptic feedback, and input type on the frequency of failures during the OS task. The QIC and QICC goodness of fit values for the GEE model were *136.518*, and *138.584* respectively. The overall descriptive values were $(M = 0.950,\ SD = 1.341)$.

No statistically significant interactions were observed between any of the main effects at any interaction level $(p > 0.05)$.

4.4.2.9 Task Load Index Ratings

Task load index ratings were recorded after each task, with the participant choosing their ratings from a 7 point Likert scale (0 being best, 6 worst). These ratings were modelled after the NASA TLX ratings, which questioned the participant on their opinions of the tasks required mental capacity, the amount of physical strain, how the task affected their pace and timing, how successful their performance was, the overall amount of required effort needed, and finally their personal level of frustration with the task.

Task load index ratings overall values were $(gM = 2.496,\ gSD = 1.227,\ Mdn = 2.5,\ IQR = 0.833,\ LQ = 2.167,\ UQ = 3.0)$, with a positive skew *(0.355)* for the mean values and the individual factors *(0.241)*.

As with the BS task, we chose two different methods of analysis for the ratings. The first being where all the TLX ratings were averaged per individual participant trials. Then secondly by comparing the individual TLX factors as a main effect. This results in a first set of analysis between haptic condition, input type, trial repetition, and then with TLX factor during the second set.

The mean TLX values reported no 3-way or 2-way significance. The haptic condition reported significant differences $F\ (1, 16) = 5.032\ p = 0.039$, where the addition of haptics increased the overall workload $(gM = 2.542,\ gSD = 1.225,\ Mdn = 2.5,\ IQR = 0.708,\ LQ = 2.292,\ UQ = 3.0)$ compared to without haptics $(gM = 2.45,\ gSD = 1.229,\ Mdn = 2.333,\ IQR = 0.667,\ LQ = 2.167,\ UQ = 2.833)$. No other main effect showed any significance.

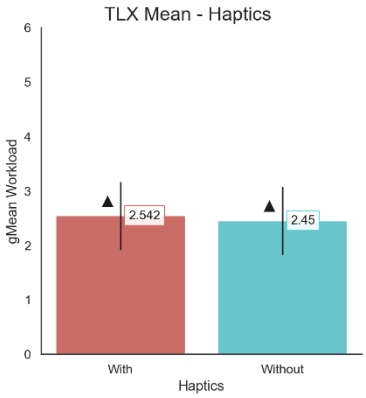


*Figure 77 Graph showing the geometric mean and geometric standard deviation of mean TLX values for the OS task, with pairwise comparison grouped by haptic application. ▲ denotes significant pairwise interaction.*Individual TLX comparisons reported no 4-way or 3-way significance. Significance was reported between the input type and the TLX factor $F\ (5,\ 80) = 5.588\ p = < 0.001$. Follow-up pairwise t-tests reported multiple significant interactions. Key interactions showed that: controllers *(gM = 2.008, gSD = 1.804, Mdn = 2.0, IQR = 2.0, LQ = 1.0, UQ = 3.0)* required less effort than hand tracking *(gM = 2.689, gSD = 1.726, Mdn = 3.0, IQR = 2.25, LQ = 1.75, UQ = 4.0)*, hand tracking had significantly lower frustration *(gM = 3.637, gSD = 1.645, Mdn = 4.0, IQR = 2.0, LQ = 3.0, UQ = 5.0)* values than controllers *(gM = 4.845, gSD = 1.309, Mdn = 5.0, IQR = 2.0, LQ = 4.0, UQ = 6.0)*, and the perceived performance was better using hand tracking *(gM = 3.694, gSD = 1.509, Mdn = 4.0, IQR = 2.0, LQ = 3.0, UQ = 5.0)* than controllers *(gM = 4.665, gSD = 1.371, Mdn = 5.0, IQR = 2.0, LQ = 4.0, UQ = 6.0)*.

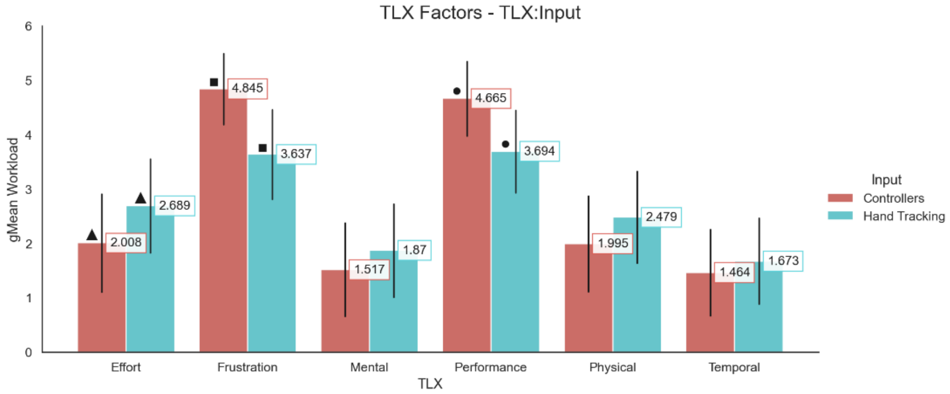


Figure 78 Graph showing the geometric mean and geometric standard deviation of mean TLX values for the OS task, with pairwise comparison grouped by TLX question category and input type. ▲■● denotes significant pairwise interaction.

Significant interactions were also reported between the TLX factor and the trial repetition, $F\ (5,\ 80) = 5.219\ p = < 0.001$, however follow-up pairwise t-tests did not report any statistically significant interactions. The main effect of TLX factor was reported as being significant, $F\ (5,\ 80) = 43.381\ p = < 0.001$, with multiple pairwise effects being reported as shown in Figure 79.

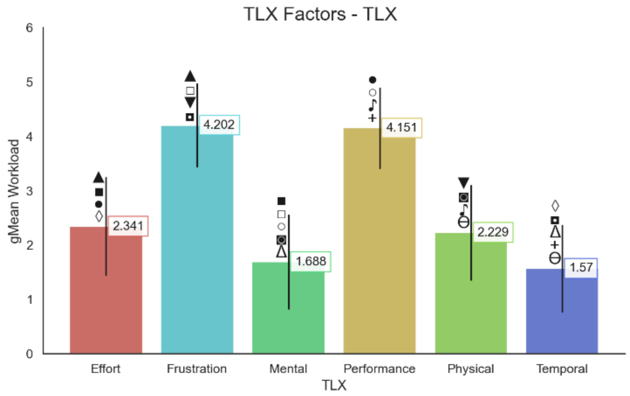


Figure 79 Graph showing the geometric mean and geometric standard deviation of individual TLX values for the OS task, with pairwise comparison grouped by the different TLX question categories. ▲■●◊▼◘○◙Δ♪+Ө denotes significant pairwise interactions.

4.4.3 Post-Study Survey

The post-study survey was primarily about finding more qualitative opinions about the tasks that the data analysis or recordings may not have directly shown. Personal comments are summarised into the two input methods, and the two different haptic effects. Two main quantitative questions were also asked during this, regarding the effect the input had on the participant's experience and the effect of haptics to both input methods.

Participants found controllers to be quite laborious having to continuously hold down buttons, or that they required exaggerated movements compared to hand tracking. Contrasting this though, there were multiple comments reporting that the controllers provided clear, smooth, steady, and consistent movements. This was coupled with a number of participants commenting they were able to more easily understand where their fingers had to be to trigger successful grabs. Overall visual representation was noticed as being less appealing and less realistic of what the participant was trying to achieve. This was echoed by several participants commenting that the speed at which fingers moved to their "interacting" state was too fast, and thus distracting.

Hand tracking issues revolved primarily around limitations of the technology; however, several were critical of the overall simulation technology and interaction logic. Jitter and inconsistencies in hand movement were reported, where for example hands may occasionally spin a full 360 degrees unintentionally, thus breaking immersion for a number of users. A few participants found it harder to interact with them due to blocks sticking to their hands, coupled with their doubtful reliability for it to perform their desired actions. One participant commented wished that they were able to roll the objects around in their hand, due to finding a continuous grasp particularly challenging.

Controller haptics comments were mildly positive, however, not significantly specifying that they were integral to the experience. A few participants commented that it was harder to discern whether they were grasping the object without haptics, something that they did not feel was an issue for hand tracking.

Overall mid-air haptics comments were neither particularly positive nor negative overall. Participants were aware of the lack of the haptics once they had received them prior, however, did not feel like they particularly added to the experience once they returned. Several participants expressed that the haptics caused them to feel that they were distracted or that they were being rushed by their addition.

Regarding the two quantitative questions asked to the user. The first being about input method, where participants were asked to rate how natural, precise, easy to use, and the visuals were represented for the input device. The second being how the addition of haptics affected interaction effects when touching, moving, grasping, releasing, the overall task completion, and the interaction with the continue button between study stages.

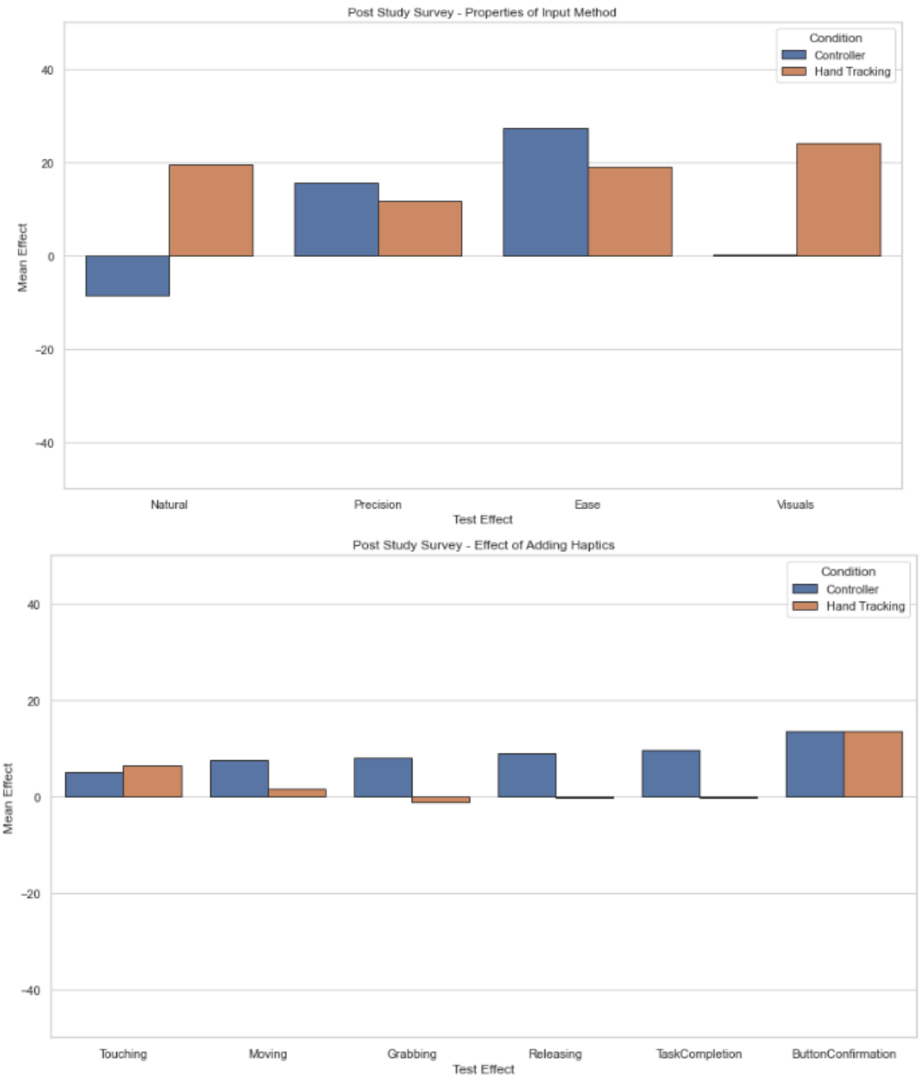


Figure 80 Graphs showing the properties that were changed with the input method (top) and the effect of adding haptics to the tasks (bottom).

From the graph above we can see that controllers were more favourable when it came to precision and ease of use. Comparatively though, hand tracking was reported to be significantly more natural and better represented visually.

Compared to differences between the input methods, the differences in haptic effect was less significant. Overall perception of haptics was positive across the board; however, controllers were significantly favoured across categories compared to mid-air hand tracking haptics.

4.4.4 Unexpected Successes and Failures

This section covers information observed during the recordings, beyond what was expected of the task.

A small percentage of participants would stand up and move around the block during the object sorting task, before grasping it and instantly sorting the object into the correct container. Doing this meant the participant would already have all the required information without having to articulate their hand. This happened both during controllers and hands based inputs, however, was more prevalent during the hand based conditions.

A very small percentage of participants would attempt to throw their blocks into the desired positions or containers, during both tasks, resulting in a mixed set of completions. A small number of participants were able to consistently complete the task by doing this, however, most were unsuccessful. These participants were never shown to attempt this when using the VR controllers, only when utilising the hand-based conditions. Similarly, a very small percentage of participants would try to grasp and move multiple blocks during the block stacking task, instantly causing failures due to knocking incorrect blocks. Coinciding with this, a small number of participants were witnessed to always fail at grasping the block on their first attempt at grasping. Failures associated with these were still included within data analysis.

4.5 Analysis & Discussion

As stated throughout this chapter, we were exploring the differences between input and haptic modalities when tasked with simple object manipulation tasks within VR. Our overall hypotheses summarised that: controllers should perform better than hand tracking when tasked with continuous, prolonged movements, and thus produce objectively better results during the object sorting task; that hand tracking should provide a more natural and expected approach to interaction, being easier to achieve desired effects, and thus should produce objectively better results during the block stacking task due to the varying object sizes and accuracy required; and finally the addition of haptics shove provide a beneficial overall effect to the participant, both in regards to task and overall study performance, and the opinions and feedback received from them.

While our study utilises highly recent technologies and devices, there are a number of limiting factors that have been mitigated as much as possible. Firstly, the impact of COVID-19 meant that our sample size was reduced significantly. Initial plans had included testing across multiple sites and with multiple populations of participants but resulted in a singular group of testers. Secondly, the methods of interacting successfully with controllers and hands, while similar in execution, were not the most common of types, especially for that of controllers. Changing the study's interaction methods may result in significant differences in user opinions and performance statistics, however, is beyond the scope of this study. This can be coupled with an ever-present issue that as newer technologies and toolkits are developed, they will ultimately try to and undoubtably will improve on the interactions that are currently available to developers and users.

As we can see from the results, there were a multitude of areas where different input and haptic combinations excelled or underperformed. Several parts of our hypothesis were correct, however there were key specific areas where this was not the case.

4.5.1 Input Modalities

Firstly, when it came to grasp times, it was clear from the data that it was easier for participants to grasp the objects when using hand tracking. There were multiple participants who, witnessed during data recordings playback, would have to manoeuvre closer to the object using the controllers, with a timid or cautious approach. It appeared that the overall lack of finite movement from the controllers was contributing to this behaviour, where participants were trying to compensate for the "binary" style finger movements This was reflected in the statistics, across multiple significant interactions, throughout both tasks, and within numerous different pairwise combinations. Recordings reflected similar struggles, with a number of participants being unable to easily grasp the blocks on their first attempt when using controllers, generally having to repeat their whole set of motions. These issues were considerably less prevalent when using hand tracking. Intriguingly, the addition of haptics provided no substantial beneficial or negative effects to participants when trying to grasp the objects.

Object sizes had less of an effect on the grasping time than expected. Hand tracking was expected to be faster in this regard, due to the heightened dynamic nature of the movement of the fingers. However, it only showed significant interactions between the largest and smallest block during the second block for the hand tracking, and the largest and medium block during the second and first trial for controllers against hands respectively. Recordings showed that the differences were often not directly related to size of the block, but the stages of the study. For the largest block it was the first time that participants were stacking the block, meaning it was sometimes used as an "adjustment" phase, with the participant recalibrating themselves to the controllers each time they swapped between input methods.

Significant interactions were observed when comparing the grasp times during the block stacking task between trials and block sizes, where grasp times reduced in geometric mean during the first trial over block size, and then increase during the second. This could be attributed to a learning effect between the two trials, especially considering that all times geometric means were equal or lower in the second trial. The evidence backs up this claim between the smallest block during the first trial and significant interactions between every single other block in both trials, where the geometric mean may have been slower, but the interquartile range was significantly broader.

From this we can summarise that it was easier and faster to pick up objects when utilising hand tracking, over controllers. Smaller objects were a definite pain point for controllers, however, were not presented as a particularly large challenge for hand tracking.

Release times showed the direct opposite trend when compared to grasp times. Hands were shown to be consistently slower than controllers when looking at the statistics, although the overall differences were smaller than grasping times. While there was a small difference between the trials in the block stacking task, it once again had less of an effect than the grasping times. However, comments about sticky objects that were challenging to release when using hand tracking, were reported by multiple participants. This was in stark contrast to very few comments about any grasping issues with controllers. These comments were also verified within the recordings where participants were shown to be placing objects in their desired location, only to have them move slightly during the release process, even in cases of incredibly slow release times. Due to the nature of how the hand tracking object interaction calculations operate, at this point in time it meant that this type of interaction was almost unavoidable without further development.

The object sorting task's lack of required precision helped contribute to an overall faster set of release times compared to the block stacking task. Multiple participants were witnessed attempting to throw blocks into the containers, while an overall trend of being more cautious when utilising hand tracking was observed. Hand tracking would often result in the participants placing the blocks directly into the container, while controllers would position themselves above and simply drop the object into the container. There was no direct secondary physics constraint, with no other block to be knocked over, within the task so participants were less careful with their movements.

From this we can discern that the process of releasing objects was more challenging when using hand tracking, with the controllers reporting times of $0.467gM$ and $0.25gM$ for the block stacking (Figure 51) and object sorting (Figure 73) tasks respectively. This being in contrast to hand tracking's times of $0.617gM$ and $0.5gM$ for the block stacking and object sorting tasks.

Hand dynamics were expected to be different between input methods, and this was realised within the results. Participants were consistently shown to be more reactive to object sizes when utilising hand tracking, with hand poses adjusting to match the variance in object size. The initial largest blocks in the block stacking task would mostly be grabbed using a whole handed, fist style approach. This transitioned with the middle block, which was mostly grasped using a multi-finger pose. Then finally the smallest block was grasped using a majority of multi-finger poses, but with a significant increase in pinches. As expected, the differences in grasp pose were not replicated through the object sorting task, with no significant interactions being reported.

From these results and data, we can see a slight difference that hand tracking showed a more dynamic and reactive approach to object interaction compared to the controllers. These adjustments may have been critical in the participants performance results when it came to interacting with smaller objects.

Failure rates were reported to be significantly higher when utilising controllers during the block stacking task, with an increased mean from 0.65 to 0.89 times per block compared to hand tracking. Multiple participants were observed during the recordings to struggle at initially grasping the smallest block during the removal stage when utilising controllers, either knocking the tower or missing the grasp entirely. This continues to be proven, with the reported failure rates for the smallest block being higher than any other block (1.17 times per block). When compared with the overall lack of adjustment in grasp pose of the controllers, these numbers make more sense as the controllers were not able to effectively adjust their positions as easily as the hands, and in turn not able to efficiently grasp the objects.

4.5.2 Tasks

The block stacking task results presented an interesting juxtaposition, where the initial hypothesis suggested that hand tracking should be better than controllers. In practice this was only partially true, with the different input methods having different strengths and weaknesses at different metrics and opinions. Primarily, overall failure rates were generally lower when either input method did not have any haptic feedback, however, differences were relatively minimal.

When we break the information down into the varying sizes, we start to see significant differences, where small blocks were significantly harder to action successfully. Explanations for this could come down to how the blocks were presented during these two distinct phases. During the stacking segment, the blocks were presented to the participant in a singular, floating, repeated position in 3D space at which they were to grasp the block from. Comparatively, the dismantling stage required far greater physical precision to grasp the block from a unique, participant created position. This was combined with the secondary issue of causing a failure state by knocking the tower, undoing the progress of that block. Both of these issues favoured a dynamic approach to the block grasping, possible with hand tracking but less so with controllers. A large portion of recordings would show the controller using participant trying to rush toward the block and quickly grasp and remove the block, attempting to reduce possible physics complications.

This difference in failure rates was surprisingly contrasted with the stacking accuracy between blocks being worse when utilising hand tracking, although this can also be contributed to the observed release differences between the input modalities. While controller results would generally place the object into the desired place, with no real difference between the intended and resulting placement position, hand tracking would often result in participants accidentally applying secondary movements. Their initial placement position being one position, and then small movements being applied to the object on release of the fingers. This was due to the system trying to calculate whether the object had been released, while still attempting to move the object relative to the hand. Final opinions of participants coincided with these observed issues, with several participants expressing that objects felt sticky, making it harder to let go of them.

One key interesting result from the block stacking task was when it came to participants workload questions. Even though the overall input methods were relatively different in a number of areas, specifically dynamics of finger positions, the resulting values for each TLX question were fairly similar for the block stacking task. Interactions were observed between the trial repetition and input method, yet once again these were relatively similar overall. The controllers reduced their workload in the second trial, while hand tracking's workload increased, with statistical interactions being reported directly between the different inputs in their respective trials and the different trials and their respective inputs. Overall, the general difference in workload between factors was relatively low for the block stacking task, which was in stark contrast to the object sorting task.

The object sorting task reported significantly higher values for the frustration and performance factors, and required nearly double the overall workload than the block stacking task ($1.264gM$ vs $2.496gM$, Figure 79). Significant effects were observed when comparing the TLX factors against the input method, where hand tracking showed less frustration and better perceived performance, while controllers required less effort. While hand tracking may have reported less frustration than controllers, this was not reflected within the post-study surveys. Multiple participants expressed annoyances at accidentally dropping blocks when rotating their hands, and thus causing a failure. While the failure occurrences may not have reported significant differences, the combination of overall task time, travel rotation, and travel distance, coupled with the recording observations, may provide some insight. Controllers were shown to have significantly higher rotational values then hands per block ($450.321gM$ vs $321.082gM$, Figure 76), had significantly larger travel distances ($80.574gM$ vs $55.966gM$, Figure 75), all while having significantly faster task completion times ($42.792gM$ vs $50.45gM$, Figure 68). Not only were controllers therefore able to achieve more than the hands, but they were also able to do so in less time. Recordings reflected this, where several participants were shown to simply avoid holding the block to find the information when using hand tracking, resorting to standing up and moving around to visually see the spot counts. The lack of failure occurrence significance could be attested to the participants who were observed performing said grasp avoidance, where they would fail on the first block and simply refuse to perform the grasping action for each block after the fact. Several participants commented that the hand tracking did not present them with consistent enough results and thus would opt for the most reliable approach, even if that meant not directly picking up the block.

Object sorting results proved our initial hypothesis, where we expected the controllers to perform better at this task. The continuous movement, and rotations required pushed the hand tracking in the majority of instances, with many participants encountering unintended positions and rotations of their hands during the task. We were not expecting participants to complete the task with hand tracking by avoiding grasping the block entirely. If we were to adjust the study to prevent this, then we would expect the failure rates and overall gap in performance to be exaggerated.

4.5.3 Haptics

The effect of haptics were less favourable than expected, in almost every category. While the initial effect of their inclusion was a pleasant experience for most participants, as several people expressed during the acclimatisation phase, those feelings quickly changed once the task requirements were taken into account. Key areas of interest for the tasks were that of failure rates, workload ratings, and grasp times, as these gave a quick insight into the overall level of challenge that the participant encountered. For almost all instances of these statistics, the haptics version of the input modality was rated less favourably, with either higher workload, an increased failure rate, or longer times. Only a few statistics were shown to have significant improvements when haptics were applied.

Workloads when haptics were applied were worse within both tasks, with greater differences being present in the block stacking task (Figure 64) than object sorting (Figure 77). This was repeated within the post-study survey comments, where haptics had an overall negative standpoint, making several participants feel rushed or agitated. As the tasks had no inherent time limit this was a surprising piece of information. While the object sorting may have not reported significant differences in failure rates as a whole, we saw significant trends within the block stacking task. The addition of haptics resulted in higher failures than without, going from an average of *0.62* per block without to *0.92* with.

The grasp time of objects were impacted in varying ways, especially when comparing by input modality. When it came to controllers, they were consistently slower at grasping objects when haptics were applied, across both tasks and between trials. However, this did not directly apply to hand tracking, where the grasping times showed similar values across trial repetitions and tasks. Interestingly, when the block size decreased, the controllers showed longer times and wider deviations of values (Figure 47), yet hands did not display this trend. While block size had a general impact on the grasp time, where the smallest block would be slower and of greater deviation, this trend was only mirrored when the haptics were applied, yet was not without. The addition of haptics appears to have simply exacerbated issues with grasping times, rather than helping to understand the point of interaction. This directly went against our intentions with the implementation. Several participants had reported an improved understanding of interacting with the controllers, where they found it easy to understand the point of interaction with the blocks. Unfortunately, the statistics directly counter this in both tasks (Figure 47, Figure 69).

Rather surprisingly, as with the difference in input methods, haptics appeared to have a similar contrasting effect on release times compared to grasp times. The release times were marginally slower during the block stacking task with haptics (1.064 times, Figure 50), however, were significantly faster during the object sorting task (0.577 times Figure 72). This was virtually the only area that the addition of haptics provided a beneficial performance benefit for either input method or task.

As the haptic implementation was designed to produce a consistent feeling of touch, the constant feedback meant that participants may have been receiving a certain level of sensory overload. While the initial helpful information of contact confirmation may have been beneficial to understand when the participant was touching and within the regions of the block, beyond that it appeared to be detrimental to their performance.

Interestingly, when looking deeper into the application of haptics, we can see possible reasons as to why differences were more apparent with controllers than hand tracking.

Mid-air haptics are applied from a position beneath the participants' palm, and thus they are not consistently felt if the participant were to invert their palm direction. This would mean in circumstances where the participant has grabbed an object, rotated it around over a number of axes, and then released, there would be a significant portion of time where the back of their hand was facing the mid-air haptic array, and thus not receiving mid-air haptics. Comments about lack of haptics during the study most likely stem from this technical and physical limitation, however, it may have resulted in better performance for the participant without being directly realised. Helping to back this assumption up is the fact that the controller based haptics did not show any improvement compared to their haptic-less counterpart. The controllers did not have the physical limitations of the haptic array, allowing for them to receive haptics at all points throughout their contact.

From these results we can confirm that haptics, in their current form were detrimental to the overall task. The effect they provided was generally too strong that it became distracting, thus harming their performance. There appears to be beneficial possibilities around reducing the amount of haptics applied to the participant, with slight gains being found within earlier trials for certain participants when it came to both opinion and overall performance.

4.6 Conclusion

In this chapter we have presented the development and outcomes from a user study aimed at exploring the differences in object interactions between novel input and haptic modalities, compared to more conventional approaches. These input modalities were chosen based upon both prior research, and current trends into the field of computer vision based hand tracking. We developed novel approaches beyond what was achievable with mid-air haptics before, while delivering a consistent VR testing environment. This allowed participants to interact and manipulate virtual blocks, under the focus of two distinct fine motor skill tasks. Our development and task choices were heavily grounded within real world developmental studies, ensuring a high level of data congruency, while being combined with novel and futuristic technologies. We implemented an experimental recording framework, allowing us to reproduce participants experiences for heightened data analysis. The quantitative data from said recordings was analysed alongside of qualitative data from in-simulation questions and post-simulation surveys, allowing us to gain a deeper understanding of the participants and their actions. Finally, we discuss the trends and results of the data, explaining the benefits and negatives of the different input methods, and their haptic counterparts.

5 PlayRecorder - The Development and Implementation of a Real-Time User Study Recording and Recreation Toolkit for Data Analysis

5.1 Introduction

Data collection and analysis is paramount to good research, without it the final result will never reach the potential that was initially set out to be achieved. Implementing cohesive systems that positively benefit with this collection can often be significantly time consuming and a large undertaking within a project, especially when working with real-time data. Within the Fine Motor Skills (FMS) project of the previous chapter, I designed, developed, and implemented a flexible, expandable, and easily implementable data recording and analysis system. This resulting system is an in-simulation data recording tool called PlayRecorder, which has been subsequently open-sourced via GitHub (Clark 2021) and is freely available to use within the Unity game engine.

Unlike a standard video recording format, PlayRecorder records the movements and changes of object logic, affecting the core underlying Unity logic to replicate the data. Researchers can setup a scene to be recorded within Unity, record participants trials, and then replay the data within the same original scene without having to build a secondary analysis scene. It links directly into the scene that was recorded, sitting atop what has already been developed rather than copying any foundational logic or objects, thus reducing the required storage requirements as it does not have to store any models, textures, or other media. The system is designed with performance and rapid expansion in mind, with a large focus on event based programming, inheritance, and threading, while retaining the ability to be easily deployed in built applications and studies.

Prior implementations of similar principles tend to either focus solely on pure text based recordings (Brookes 2021) or video (Unity 2021a). Other similar implementations are either paid (Cognitive3D 2021), or are designed for generic large scale user analytics (Unity 2021b) instead of singular individual simulation recreation. Comparisons between prior user study recording setups have shown significant increases in data quality, reductions in file size, and heightened ease of use and re-analysis. This is coupled with a significantly easier workflow for obtaining and producing statistics post-study compared to video formats.

5.1.1 The Problems

Many current data collection and analysis tools available can generally be summarised as one of two types of implementations. Either they present an entirely visual approach where they record video files from either the participant viewpoint, a secondary viewpoint, or from multiple camera feeds, usually via a tool that is external to the simulation. Or they opt for recording the statistical information from the study into data formats such as CSV (comma separated values) or JSON (JavaScript object notation), performed from directly within the simulation.

5.1.1.1 Video Recordings

Video recordings allow for the pure visual output of the user to be recorded, allowing for quick visual analysis of interactions and study progress. Common recording setups tend to show either what the participant was seeing directly, or a secondary viewpoint from which they were observed. Both of these setups can be used to give different visual insights into the participants experience with a study. While these are great at giving quick insights, they leave a lot to be desired when it comes to a number of areas such as high performance overheads when visual quality is required, lack of true quantitative data collection, and cumbersome multi-user analysis.

Recording video of the simulation introduces a performance overhead that is often overlooked during study development. Including a video recording system within a simulation framework generally directly affects performance with the underlying logic, thus negatively impacting the user experience and to that extension, user data. With XR being increasingly common within HCI studies, the available bandwidth for extra services, be it processing headroom on the CPU or overall available RAM, is significantly reduced, especially when frame rates of 90Hz or higher are required for the simulation itself, let alone any recording. Coupled with this, video files are generally larger when compared to other data types (e.g., 100mb plus) such as text-based statistics or databases, which can significantly increase the amount of storage required for the study data. Videos can be anywhere from 2.5 megabits per second (Mbps) for 1280 by 720 pixel resolution, 30 frames per second YouTube videos (YouTube 2021), which are highly compressed and unlikely to be the size of the raw recording, all the way to 663Mbps for uncompressed 720p 30fps video (Wikipedia 2021e). Screen recording tools such as Sharex (Sharex 2021) and OBS (OBS 2021) require a relatively extensive amount of setup for the user, often create large file size recordings, and are at times restricted to windowed only applications.

Video recordings are entirely frame rate restricted, which at times can cause issues when recording certain types of information. For example, if a study was measuring the reaction times of a participant in the millisecond ranges within virtual reality, large variances in information would be quickly lost if recording a video at 30 frames per second compared to the 90Hz of the VR headset. The possible ranges would be reduced 3x and quite possibly even miss the frame of action due to the low overall speed. Similar effects could be witnessed when trying to understand task completion times where the participants are aiming for speed, with overall variances of times being quantised.

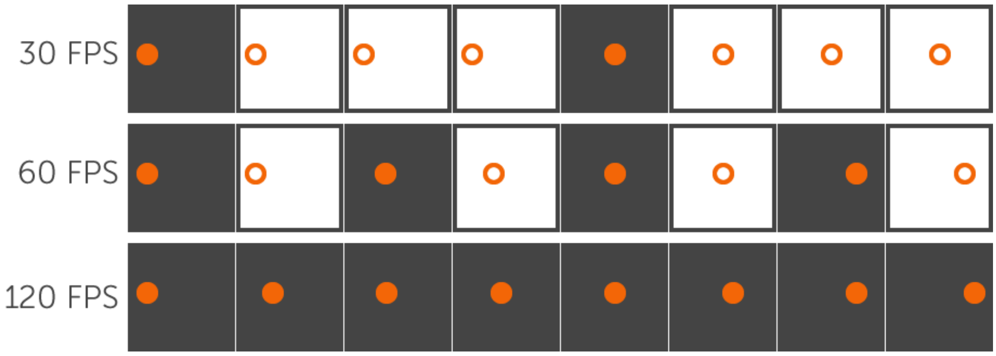


Figure 81 An example of the differences in frame rates with a constant moving circle. Outlined frames represent the lack of an update and the resulting repeated image. At 30 frames per second the circle in the image is only shown to change from the left to the middle side of the frame. At 60 frames we have twice as much information and can start to see the resulting position. Finally at 120 frames we see the full translation of the circle from start to finish.

The reliance on a single perspective has made video a challenging tool throughout many an application, especially when using it for analysis of intricate structures and movements, such as a hand. As the video is a recorded frame, there is no possibility to adjust the selection of items in the frame, nor can a perspective be changed after recording. Not only is it impossible for the camera to capture information from every angle, but it is also often not entirely feasible to ensure that the camera will have a wide enough field of view to capture all information throughout a study. Using a field of view too narrow will clip large amounts of information, while too wide and the recording will appear far too zoomed out and distorted. While a camera could be moved further away using a narrower field of view, this can often introduce

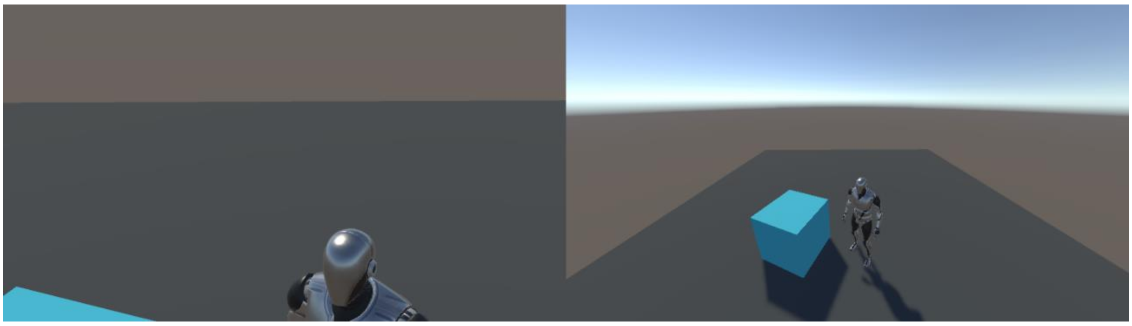


Figure 82 An example of two different fields of view, the left being 40 degrees horizontally, and the one on the right being 120 degrees. The left is too close and makes it impossible to see the full extents of the character and object, while the right is too far zoomed out while also introducing distortion and thus hard to see all detail.

Analysing video recordings can be incredibly challenging, and generally requires an entirely manual approach. This can lead to two core problems. Firstly, the researcher will be required to manually analyse each recording in a similar fashion, which is incredibly challenging to ensure consistency between recordings. Secondly, the research may attempt to implement a computer vision based approach to the data analysis, which would have to be developed specifically for the study, using tools such as OpenCV (OpenCV 2021) or Tensorflow (TensorFlow 2021). Both of these options are highly time consuming, cumbersome, and generally unrealistic for most studies. Computer vision approaches may eventually be a feasible approach, but would require significant quantities of data, beyond what many academic or small scale business research tend to collect.

Many researchers make use of the Unity (Unity 2021c) game engine for its flexible 3D development environment, making it easy to produce simulations within many different computing and reality platforms. The large market share, especially within the research populace, makes Unity a logical choice for development of the tool. Currently available in-engine tools that allow for recording visual information, such as the Unity Recorder (Unity 2021a), are generally limited to only the editor when recording footage, or only output video based formats. Running applications in the editor introduces performance overheads that are not experienced during built applications, coupled with overall reduced portability. Many of these current tools are entirely focused on the creation of content, rather than analytical information gathering, meaning whatever system we would be using would not be ideal for the task. This would likely result in considerable effort during implementation or require numerous changes for future work. External screen recording tools, such as Sharex or OBS, offer flexible methods for recording video, generally allowing for custom resolutions and different levels of frame rate, however, cannot make use of any in-engine logic. This prevents them from being able to hook into any events, such as trial beginnings and ends, making it more cumbersome when trying to automate recordings. External recording devices such as video cameras or external capture cards do alleviate the issues with performance, however, they still inherit similar problems surrounding timings and external factors. Video cameras recording screens are a viable option in certain scenarios, but would not be feasible in scenarios where the participant is wearing a VR or XR headset.

5.1.1.2 Text-based Recordings

Statistic text-based recording methods can be used to capture data from studies, allowing for quick exporting into other analysis environments such as Python (Python 2021), R (R 2021), or SPSS (IBM 2021). These formats such as CSV and JSON are easily written, read, and modified, allowing for highly flexible data analysis post-study, while being simple to implement and maintaining a low performance impact on the simulation. Numerous data analysis tools within Python, such as Jupyter (Jupyter 2021) and pandas (pandas 2021), make full use of CSVs due to their overall speed and lack of complexity when processing. There is relatively little effort required to collect these variables, with the majority of the work tending to stem from the process of initially generating them. As many of these formats are entirely text based, with a few small bits of formatting, it makes them very easy to preview and inspect.

While perfect for gathering and measuring quantitative and text based data, text-based recordings lack the visual benefits that video recordings provide. This can make it challenging to understand entirely what happened, or what caused specific values to be recorded. For example, the time taken before grabbing any object may be recorded, however, it would be challenging to understand why the participant had failed at certain task completions without the visual data to accompany it. Subsequently, due to their lightweight nature and structure, these formats can often be particularly fragile when recording and modifying data. Incorrectly modifying just one comma in a CSV can offset or corrupt the entire file, which can be incredibly time consuming and challenging to rectify.

5.1.1.3 Common Issues

Although both of the described methods provide workflows for capturing data during the study, neither allow for easy ways to generate or capture new data post-study. This means the collected data is in its final form, with large amounts of work required to garner new material from it. Due to the lack of this flexibility, user studies are more challenging or simply unfeasible to be re-analysed in future. This can create instances where key information is overlooked, with the final output from the study being factually incorrect, even if the reported results and statistical analysis reports significance.

Very few current toolkits offer solutions that encompass both, or multiple, methods of data recording, focusing on either video recording or text based solutions, and ones that do can often be costly such as Cognitive3D (Cognitive3D 2021). Implementing both can often be cumbersome and challenging for the researcher. This accidental complacency can add considerable extra work onto the development of a study, which the researcher may not have initially planned for. Due to this it often results in less than ideal situations where one solution may end up being chosen over the other, or a secondary amalgamation of less than ideal technologies. From these issues we could clearly see several problems that were in need of being solved, while also providing the opportunity to integrate knowledge and ideas from the realms of data analysis rather than simply just data collection.

5.1.2 The Requirements

Based on the problems with current toolkits, we realised there were a number of requirements that were needed to be fulfilled. Whatever toolkit we are going to develop needs to successfully achieve our intended requirements, while still being flexible enough to expand for future projects. This means our tool needs to be structured and must operate in a certain set of ways.

1. The system should allow for customisable and flexible recreations of simulations.

2. All data should be flexible enough to expand in future, without the system directly being made to understand it each time new items and logic were added.
3. The toolkit should be performant and lightweight within any researcher's existing simulation.
4. The system should be easy to use at all stages, be it implementation, playback, or expansion.
5. The developed tool should work within built applications, just as it does within the Unity editor interface.

By meeting these requirements, we can have a greater chance of reimplementation in future projects, while consistently maintaining beneficial outputs for analysts. Each of these will help ensure the tool will provide a positive and genuine benefit compared to the usage of recording pure video or logging raw text files.

5.1.2.1 Design Considerations

From these requirements, several design considerations were taken into account. These would guide the development, while also influencing future changes and developments.

5.1.2.2 Customisation and Flexibility

Ensuring that the system was customisable and flexible for the recreation of simulations meant treating the data within the simulation as distinct separate parts. To this degree, each of the elements within a scene would be treated as an independent element. For example, the input method of a user would be one piece, while the objects they were interacting with would all be individually separate ones. This also meant that we needed to allow objects to be interchangeably added and removed from the recording setup, allowing for varying ways in which a scene may be recorded. Playback should also allow for this flexibility by enabling both full scene recreation but also only specific, desired elements to be actioned. For example, if you recorded a participants input and object interactions, you should be able to hide the objects to see just their input changes.

5.1.2.3 Data Expansion

Data expansion and customisation would allow for developers and analysts to bring their own logic into the system, doing so requires a specific design method for this. The open-closed principle (Wikipedia 2021f) would suit this requirement entirely, specifically with a strong reliance on inheritance. This principle is the method of having a *"parent"* or *"base"* class which holds information and methods that are then inherited by *"child"* classes which override or extend the original declaration. This is done to allow the child class to expand on the logic of the parent, enabling specific pieces of information to be understood. These child classes can be stored in the same declarations as their parent classes, while retaining their variations, and can be converted back to the child type with ease. Using these principles would allow for simpler expansion by other developers and researchers, making it easier to tailor the features and benefits of PlayRecorder directly to their simulations.

5.1.2.4 Overall Performance

Keeping the toolkit performant and lightweight was integral to the overall usage and would directly affect the chances of it being reimplemented in future studies. This meant designing logic flows that produce minimal performance overhead and reducing reliance on the same threads as the core application. By ensuring the system has little impact on the base simulation, we can improve the possible reach of implementation, especially in scenarios such as high frame rate VR or AR applications. This also encompasses the need for data files that are small in size, which in turn would reduce the storage requirements for both the recording and analysis segments of PlayRecorder's usage.

5.1.2.5 Ease of Use

Ease of use would tie strongly with the overall performance and flexibility of the system. This meant we need to allow for data to be easily created and modified to meet custom simulation needs, with relative ease and minimal effort. While it would be impossible to ensure that all logic could be recorded without the introduction of additional scripts by the researcher, we could at least design the system to not conflict with currently implemented code and logic found within their simulation. To this degree it would mean implementing features such as custom Unity editor windows, as well as providing code hooks into engine logic. Custom editor windows allow for the creation of secondary tools and features that may fall outside of the scope of recording and playback but would stray more towards data analysis. Their addition would help reduce friction for the researcher when trying to work with their recorded data.

5.2 Methodology and Development

The developed system allows for researchers to record studies within their Unity simulation, and then replay them within the same toolset used to develop the study. It can record logic within built applications, and then loaded within the Unity editor without having to make any adjustments to the original study scene.

Conventional video based recording methods capture individual frames of visual information from a pre-determined point of view and fixed resolution. The resulting information is generally only a small portion of the actual simulation, being limited by the camera's field of view, or by simply having key pieces of information being occluded due to the participants interactions. PlayRecorder takes a different approach by recording state changes to object properties and values, with the original objects and scene being reused for playback. This means that it does not understand what the objects actually are, but simply what several of their key values are e.g., knowing the changes of a 3D model's positions and rotation, but not what the 3D model itself is. One major benefit of this approach is that we do not need to record the information of objects every frame, simply their new values and then evaluate between the old and new. This significantly reduces the overall requirements for both storage amount and processing power required.

As we are implementing the tool within Unity and utilising object property information, rather than pixel information, there is no overall resolution constraint to recordings achieved through PlayRecorder. This means that the final produced file can be likened to that of a scalable vector graphic (SVG) files instead of bitmaps, where it can be infinitely scaled up and down, without degrading the overall visual quality of the visual output. Due to the usage of the game engine, we gain the secondary benefit of having no restriction on the final viewpoint, allowing for fully adjustable positions and views throughout playback for free.

"Ticks" are repeatedly referenced throughout the system and refer to an increasing numerical value, starting from zero at the start of the recording, and declared when a frame recording is requested by the system. These are absolutely paramount to recording and understanding any recorded data as they are the single constant between all frames within the system. Multiple underlying parts of the system make use of the current tick, allowing them to quickly align themselves to the current frame. *"Ticks"*, and to that degree recording updates, are time agnostic, meaning they are separate to the frame rate of the application. The system stores three associative values to the tick in each recorded file: the timestamp from when the current recording was started, the total number of ticks recorded for the file, and the frame rate at which the file was recorded. Using these three pieces of information, we can quickly decipher how long the recording lasted and when it

started and stopped in date and time, while consistently being able to iterate through the recording with no more computation than simply increasing an integer by one.

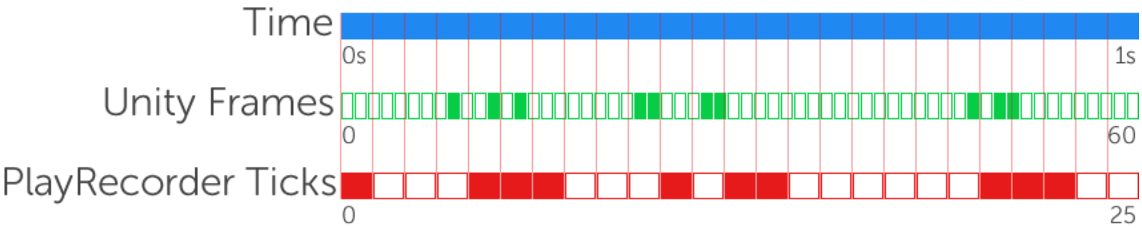


Figure 83 The diagram above visually explains the differences between frames and ticks, with time going from 0 to 1 second, a Unity frame rate of 60 frames per second, and PlayRecorder ticks occurring at 25 frames per second. Green boxes represent when a change to an object has occurred, and red filled represent when PlayRecorder stores said change. As PlayRecorder listens to complete changes, it can result in instances of a delay between change and recording or, result in multiple changes happening before recording, as shown in the 12th red box. If the PlayRecorder tick rate was equal or higher than the Unity frame rate, then it would capture every single possible change.

5.2.1 System and Data Structure

The PlayRecorder system consists of four core parts: the recorded data, the record components, the recording manager, and the playback manager. There are a number of extra tools that sit on top of these parts; however, these are the ones that are integral to the system.

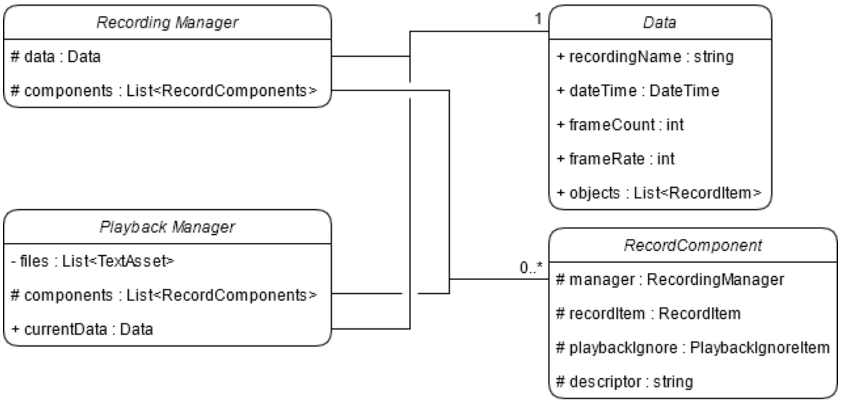


Figure 84 This class diagram shows a basic level of how the system is interconnected. As displayed, the data object is handled singularly by the recording manager, where it is created and saved, and read from saved files in the playback manager. This diagram is only showing the critical information for the system and will be expanded upon within other chapters.

Both recording and playback is achieved through the usage of secondary custom threads to the simulation, which reduces load on the core processing but in turn creates several challenges. By default, Unity is far from agreeable when it comes to using custom threads. Many of the core logic operators simply refuse access when trying to do so within threads, even to the degree of not being able to access the name of a game object from within a thread. To help circumvent these issues, a secondary set of functions that run on the main thread of the simulation were created. These functions generally perform caching queries in one of two ways. During recording, data is cached as the simulation runs, this is then stored within secondary data objects created by the recording thread. While the playback thread performs the opposite effect, by updating a set of caches that are then applied during the main thread update cycle. This method of having caches between the main thread and custom threads, a similar effect as using thread pools (Wikipedia 2021), helps to reduce memory issues and possible crashes.

5.2.1.1 Recorded Data

A key defining point of PlayRecorder is that all recorded data within the system is completely polymorphic, right the way to the root data object. This means that every part of the recorded data can be extended with custom attributes, information, and frame types, allowing for extreme amounts of flexibility. To understand how this recorded data works, we will be breaking down each segment of the structure.

Within the system, all data is agnostic to the underlying operation of the managers. Although the recording manager and playback manager may perform actions on top of the data, they could quite easily be replaced with custom implementations with relatively minimal effort. All of the data is stored as serialisations of custom C# classes, which is then saved as binary files.

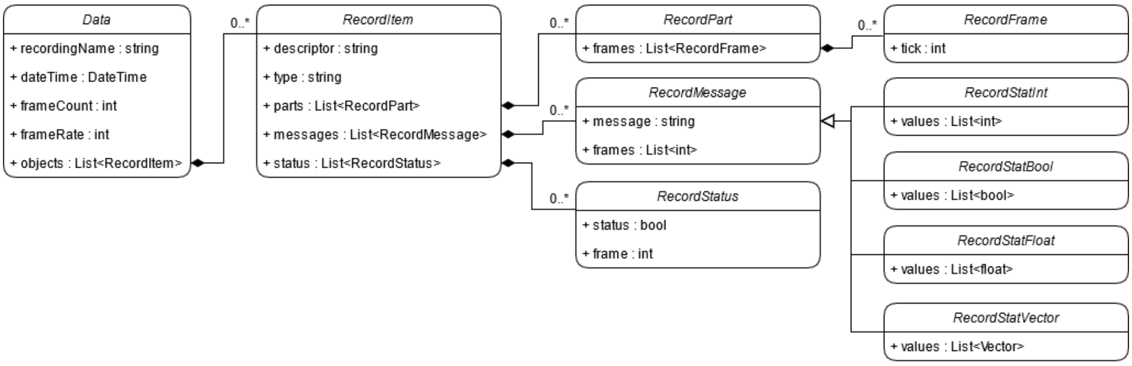


Figure 85 The class diagram above shows the outline of the data structure for each file. The "RecordStatVector" has been simplified, with options for Unity's Vector2, 3, and 4.

At the root, the *"Data"* object holds the core attributes of the recording such as: the recording name, date and time of the record, frame rate, and an array of *"Item"* objects. Each part of the data structure that can be modified is labelled in a similar fashion, where the base level item is prefixed with *"Record"* and then suffixed with what the class is. Throughout this explanation all instances of described objects are prefixed in code with *"Record"* but are described without them for the sake of conciseness.

The *"Item"* object is the root piece of information that is attached to the varying game objects within your recorded scene. These are controlled by the *"RecordComponents"* (as explained in 5.2.2, page 116) and store a unique descriptor and the type of the *"Component"* which this *"Item"* was attached. Each *"Component"* will generate an *"Item"*, meaning the final *"Data"* file will include an array of *"Items"* equal to the number of *"Components"* in the scene. Every part of the data explained beyond this sits within the *"Item"* and is contained to that specific object within the scene. There are three separate arrays of information in the "*Item*", the *"Part"*, *"Message"*, and *"Status"*.

"Part" is the object in which all the state changes, called *"Frame"*, of the *"Item"* for your overall recording are stored. The idea behind this is that an *"Item"* can have multiple parts, such as if you were recording multiple transforms of a dynamic armature (character bone structure), and due to this the parts individually record and manage their frames. This level of abstraction lets you override every chain of data easily, reducing the amount of redundant or repeated code. For example, you could implement the base *"Item"* coupled with a base *"Part"* and a second custom *"Part"*. The first base *"Part"* could have custom frames inside of it to record the position of the object. While the second custom *"Part"* could have custom information such as average velocity or simply an answer to a form, it may not need to record many, or any, frames during recording and thus does not need to have custom frames. This flexibility with the system allows for developers to efficiently

manage expected file sizes and processing, even when a large number of individual objects are being stored.

A "*Message*" is similar in principle to that of an in recording log file. *"Message"* objects are stored inside of the "*Item*" object as an array, with each *"Message"* object storing an array of ticks, specifying the ticks at which the message appeared. Basic *"Message"* objects only store the name of the message and the array of ticks, while statistic extensions include a secondary array that allows for the recording of values coupled to a tick value. This design method means that the recording will only ever store received messages, with minimal data usage as the array of ticks naturally lends itself to a small amount of data compared to storing a whole new message each time. Messages do not store any particularly complex information about the recording; however, they allow for the recording file to be quickly analysed and viewed without having to visually play through the entire file. These *"Message"* objects are used extensively through the timeline and statistics tools.

The *"Status"* object is one of the few pieces of the recording that is automatically recorded and managed and is not directly designed to be modified. It captures when the attached game object is enabled or disabled within the scene, allowing it to be correctly mimicked during playback. Underlying structure and workings of the *"Status"* object is closely shared with the *"Message"* object, however, was explicitly differentiated to ensure it would not be overwritten accidentally. While the *"Status"* could technically be attached to the frame, not every frame of information being recorded is that of a Unity game object. As each *"RecordComponent"* is meant to be

The *"Frame"* is the final piece of the structure, where all information about the changes of the recording are stored. As covered before, the *"Frame"* object only records changes, rather than constant information. This results in an array of frames that is different in size to the overall number of ticks in the recording. Each frame stores the tick at which the change was witnessed and thus allows for evaluation based on the current tick relative to the playback manager. *"Frame"* objects can hold any form of data, as long as it can be serialised.

5.2.1.2 Serialisation and Unity

There are a few issues when utilising inheritance within the Unity engine, specifically when it comes down to the serialisation of the child classes and data. Primarily, the Unity editor interface, by default, does not handle inherited data particularly well when serialising. Many instances of trying to store and load inherited data, result in the Unity editor reverting to the base classes and types. While on its own this would not be a particularly problematic issue, as we were implementing multiple custom editor interfaces for handling various this was a critical problem. Recorded files would be loaded into a Unity editor interface, see 5.2.4 Playback Manager, which would then become the central point of reference for all further handling of data. As soon as the editor would enter play mode, or perform other "reload" style functions, these files would revert any extended pieces of data to their original form. This meant any custom *"Frame"* types, *"RecordItems"*, and other key parts of the recorded data was reverted to the original format, essentially rendering them useless.

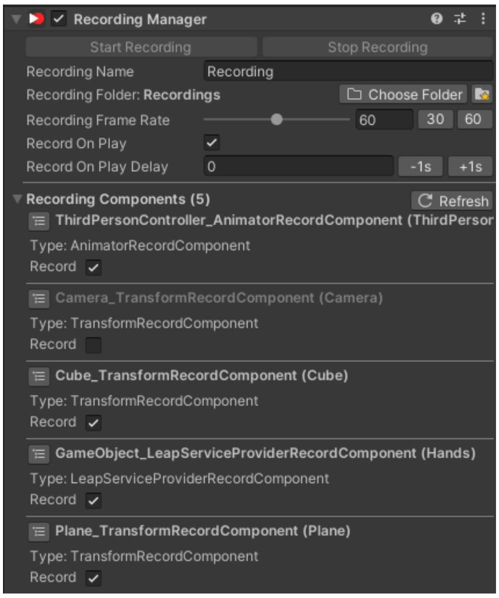
This issue extended to the default serialisation tools built into the Unity engine. While they were able to take our recorded class data and store it as binary or JSON file types, they were not able to understand any modifications or inherited data classes or types, resulting in the same problem as within the editor interfaces. To help alleviate this issues, we utilised the open source serialisation package called Odin Serializer (Sirenix 2021), which fully supports inheritance throughout the loading and saving of files. It also provides significant performance benefits compared to the built in options, while also allowing us to perform serialisation actions on separate threads. This means that we can store data without having to lock the main simulation thread of the system, which helps alleviate possible performance issues.

5.2.2 Record Components

A *"RecordComponent"* is the basic logic piece that controls the *"RecordItem"* data object. Both recording and playback logic is stored within them, with the different methods being called by their respective managers. Each *"RecordComponent"* is designated with a "*descriptor*", used to identify itself within the recordings. This descriptor is unique to the current recording setup. A secondary option designates whether the component is required for the current recording. This allows for components to lie "dormant" within the scene and only be included in a recording where needed.

As PlayRecorder is a highly event driven architecture, the component will only process updates when it receives a request to do so from a manager during recording or playback. Each individual cache is still updated, regardless of the manager events, as these are used to store the changes from the main Unity thread. This reduces the processing requirements on the individual components as they do not need to constantly check that the system is recording or playing. In turn, this also prevents issues where components may accidentally end up out of sync with the current tick. Unity's default order of execution (Unity 2021d) operates through the usage of update and fixed update cycles to determine when to perform logic. The execution of these are determined by whether the object is enabled or not within the scene. If an object is disabled it can still have its functions called but is no longer included in the overall update cycle. PlayRecorder sits at an abstracted layer above this order flow, meaning that it calls these recording and playback function for each *"RecordComponent"* irrespective of its game object state.

5.2.3 Recording Manager



☑ Recording Manager
Recording Name: Recording
Recording Folder: **Recordings**
Recording Frame Rate: 60
Record On Play: ☑
Record On Play Delay: 0

Recording Components (5)
ThirdPersonController_AnimatorRecordComponent (ThirdPersor
Type: AnimatorRecordComponent
Record ☑
Camera_TransformRecordComponent (Camera)
Type: TransformRecordComponent
Record ☐
Cube_TransformRecordComponent (Cube)
Type: TransformRecordComponent
Record ☑
GameObject_LeapServiceProviderRecordComponent (Hands)
Type: LeapServiceProviderRecordComponent
Record ☑
Plane_TransformRecordComponent (Plane)
Type: TransformRecordComponent
Record ☑

Figure 86 The recording manager within the Unity Editor. It shows all the items in the scene and lets the developer quickly adjust every setting.

The recording manager is the item responsible for managing and initiating recordings. It dynamically understands and registers all the *"RecordComponents"* within the current scene and adds them into the array of items to be recorded. Recordings can be started directly at any point through the editor interface, or automatically through code, which in turn allows it to be easily integrated into existing systems or automate the recording process.

Each of the components in the current scene can be ignored from any recording, this allows for more flexible recording setups. For example, a study may have multiple input methods being constantly present throughout the scene, but only actively used between individual trials. Rather than having to add and remove components from each of the input methods, the system can simply be told to ignore the non-active inputs and only record the required ones. Although standard Unity practices would suggest this should rely on the enable and disable events of a game object, PlayRecorder sits at a layer of abstraction above the standard logic of Unity, making this a non-viable approach. As an example, if objects were being ignored based on their object status, an object that was disabled before recording started would not be included during recording. If this object was simply not currently active within the scene, due to being a later part of a study, this important piece of information would be entirely excluded from the recording.

When recording, the manager operates through a separate thread, with each *"RecordComponent"* update being requested at each recording tick. The tick update occurs on the thread, and thus happens at a different time to the main frame rate of the application. Each *"RecordComponent"* stores the updates that happened on the main thread into a cache, and then stores them into their frames once the update is called from the recording thread by the manager. This does mean that if the recording frame rate is significantly slower than the theoretical update rate of the main thread then the differences between objects may create large jumps, as the cache only stores the most recent update. Once the recording is finished, the manager iterates through all of the recorded components, and receives their individual *"RecordItem"* objects. These are collated into one array in the data object, and then finally serialised into bytes and then stored as a file.

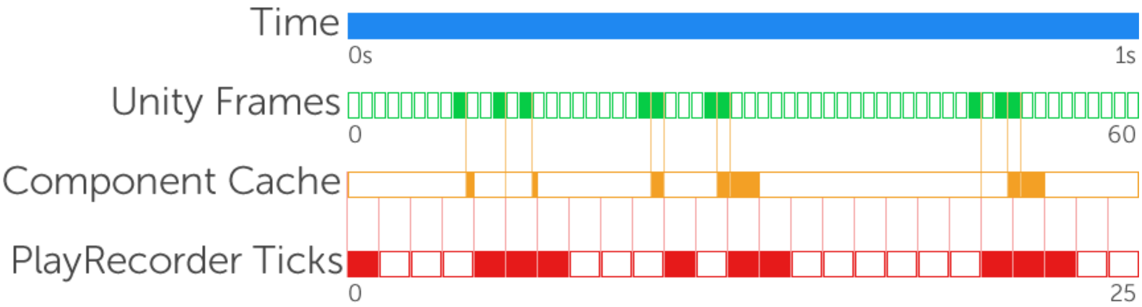
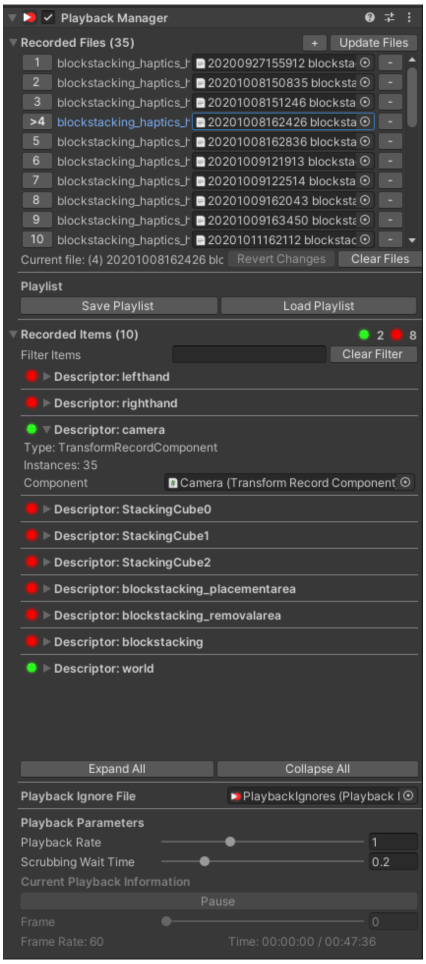


Figure 87 The above diagram expands on the prior explanation of updates with the inclusion of the component cache. Boxes in bold signify changes have occurred. As the cache does not update itself, or follow any specific timing, it simply reports whether it has been updated to allow for the recording thread logic to be called. This is the intermediary storage location for changes and occurs on the main Unity thread. PlayRecorder accesses this cache during the recording thread to then store those cached changes into a frame object.

Of the core tools, the recording manager is designed to be the least intrusive. The majority of the logic is meant to be located within the *"RecordComponents"*, rather than the recording manager, thus making it relatively lightweight. All *"RecordComponents"* are automatically added when found within the scene, thus reducing the required work for the researcher during setup. The only controls available are for setting the recording name, adjusting the recording frame rate, and whether the recording should start when Unity play mode is started.

5.2.4 Playback Manager

The playback manager is the main point of call for loading and playing files into the PlayRecorder system. It handles all playback events and allows for any number of recorded files to be brought into the system for playback.



Recorded Files (35)

1 blockstacking_haptics_h 20200927155912 blocksta
2 blockstacking_haptics_h 20201008150835 blocksta
3 blockstacking_haptics_h 20201008151246 blocksta
>4 blockstacking_haptics_h 20201008162426 blocksta
5 blockstacking_haptics_h 20201008162836 blocksta
6 blockstacking_haptics_h 20201009121913 blocksta
7 blockstacking_haptics_h 20201009122514 blocksta
8 blockstacking_haptics_h 20201009162043 blocksta
9 blockstacking_haptics_h 20201009163450 blocksta
10 blockstacking_haptics_h 20201011162112 blockstac

Current file: (4) 20201008162426 blc

Playlist

Recorded Items (10) ● 2 ● 8

Filter Items

● Descriptor: lefthand
● Descriptor: righthand
● Descriptor: camera
Type: TransformRecordComponent
Instances: 35
Component: Camera (Transform Record Component
● Descriptor: StackingCube0
● Descriptor: StackingCube1
● Descriptor: StackingCube2
● Descriptor: blockstacking_placementarea
● Descriptor: blockstacking_removalarea
● Descriptor: blockstacking
● Descriptor: world

Playback Ignore File: PlaybackIgnores (Playback I

Playback Parameters
Playback Rate: 1
Scrubbing Wait Time: 0.2
Current Playback Information
Frame: 0
Frame Rate: 60
Time: 00:00:00 / 00:47:36

Figure 88 The playback manager as present within the Unity Editor. Any number of files can be loaded, and all features are easily accessible.

All recorded files need to be loaded into the playback manager to allow for playback. Files are loaded into the cache of the system, and in turn parsed to understand the *"RecordItem"* objects located within. These *"RecordItem"* objects provide information about what type of object was recorded, as well as information regarding their original game object within the scene. If a match between the recorded descriptor and type is made, then data will be automatically assigned to the correct *"RecordComponent"*. In similar vein to how you can ignore components during recording, none of the *"RecordItem"* objects need to be assigned for playback to begin. For example, if you recorded a pair of hands as separate pieces but only wanted to view the left hand, you could simply remove the right hand *"RecordComponent"* association.

During the loading of files, the playback manager creates a cache of the data which encompasses most of the raw file information excluding individual frame data, as well as some basic information about the *"RecordItem"* objects. A global level cache of the messages and statistics is also generated, converting it from a per *"RecordItem"* level array, into a file level array and thus in turn allowing it to be easily parsed over. By creating this centralised cache of data, it allows for other pieces of code and tools to access the information from the files, without having to directly parse an entire file. The playback manager still provides access to the currently loaded files; however, the cache is often smaller in memory size for all files, compared to just a single loaded file, as the individual frame information is not provided through the cache.

Many parts of the playback manager operate directly in the inverse to that of the recording manager, especially when it comes to actually playing the recorded files. When Unity enters play mode and PlayRecorder's playback begins, the currently selected file is loaded into memory. Each of the assigned *"RecordComponents"* are sent their respective data at (the *"RecordItem"*) at the start of playback, just like how the recording manager only receives the data from them at the end of the recording. This data is then processed and actioned upon by the individual *"RecordComponents"*. A separate thread is created for playback, similar to the recording manager, where the current tick is updated and sent to the *"RecordComponents"*. The playback thread ensures time consistency across components, and the initial sending of data to each component reduces the amount of memory being sent through to each object. As the components receive this tick, they update their individual *"RecordParts"*, and update a secondary information array. This information array informs the component as to which parts have been updated during the threaded playback tick, and thus may require a change on the main thread. Without this cache based system (if playback was entirely threaded) Unity would throw numerous errors about functions being inaccessible from custom threads.

The playback manager includes a few simple, but useful, playback controls, similar to a video player such as VLC. With these the analyst can play and pause all recordings, swap between loaded files, speed up and slow down playback speed, and jump to desired sections of files. Playlists can also be created, allowing for the analyst to save the current set of loaded files into a playlist file, and then load them back in future, making it faster to swap between data sets.

Any recordings done at a low frame rate will be shown with large jumps between data, likened to that of a sub 24 frames per second video. This can at times be more prominent than video as the objects within the environment do not explicitly produce any natural or post-processing motion blur, compared to what you would find in a camera due to the lack of a shutter speed. While interpolation could be added, its addition could result in both good and bad effects to the system. Negatively, the system would have to understand and process the differences between each frame on the fly, thus introducing more overhead during playback. Coincidentally, the addition of interpolation may result in data that is harder to analyse, as frames will be estimated between states rather than specific values. This state estimation could also result in largely incorrect interpolation if the system only has information of the current and next frame, as frames are not recorded for every object at every tick. However, the overall visual appearance of the playback will be significantly more pleasant when working with lower frame rate recordings, as objects would smoothly transition between their states.

5.2.4.1 Playback Initialisation

During playback initialisation, PlayRecorder will automatically try to disable non-PlayRecorder scripts that reside on a *"RecordComponent"* object. This is done to prevent existing logic from adjusting objects during playback. Developers can override the adjustments performed to an object through the use of a "*Playback Ignore*" file, which can be specified on a whole system or individual *"RecordComponent"* level.

"*Playback Ignore*" files allow for the developer to specify which key pieces of Unity logic will be enabled or disabled, as well as any custom script names or namespaces to keep enabled. Unity logic encompasses information such as renderers (visual output of 2D and 3D graphics), collisions (for physics), and cameras. Custom components are specified through the use of their full class name (e.g. *UnityEngine.Transform*) or a part of it (e.g. *UnityEngine.*). When playback starts, the *"RecordComponent"* will gather a list of all scripts on the current object, and then iterate through them and disable every one that is not included in the list of enabled components.

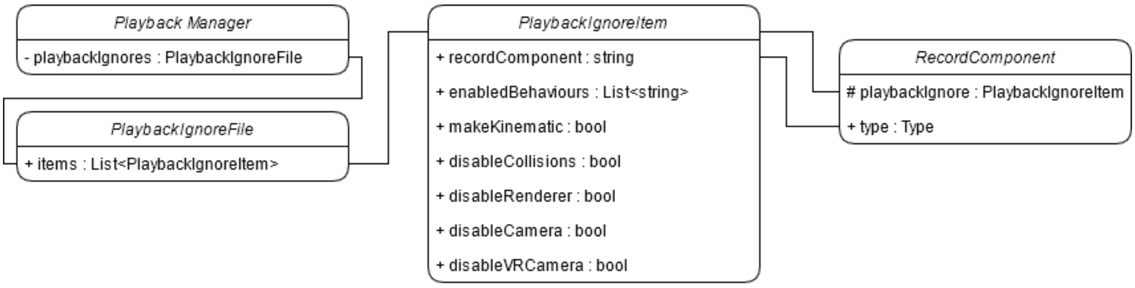
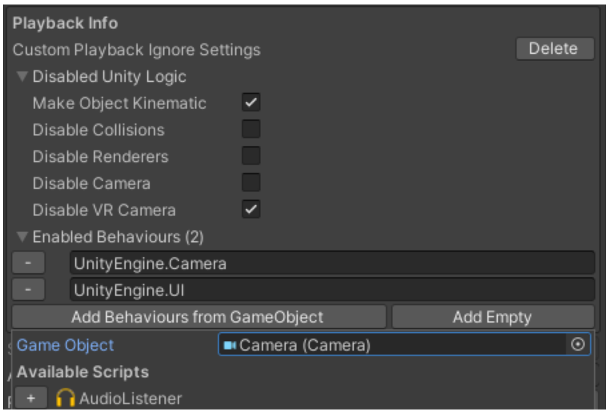


*Figure 89 This class diagram shows how the "*Playback Ignore*" system fits into the playback manager and each of the "*RecordComponents*".*



Playback Info

Custom Playback Ignore Settings

Disabled Unity Logic

Make Object Kinematic ☑

Disable Collisions ☐

Disable Renderers ☐

Disable Camera ☐

Disable VR Camera ☑

Enabled Behaviours (2)

UnityEngine.Camera

UnityEngine.UI

Game Object: Camera (Camera)

Available Scripts

AudioListener

*Figure 90 The editor interface that lets the analyst setup and adjust their customisations for each "*Playback Ignore*".*

Each *"RecordComponent"* has three different points as to where a "*Playback Ignore*" can be specified. Firstly, the default value which can be controlled through declaration within code. This allows for the developers of custom components to specify logic that they expect to be present on the *"RecordComponent"*. Secondly, the ignores can be specified as a single user created single file for the entire playback. This method generates a file stored within the project that overrides the defaults for every instance of a *"RecordComponent"* in the scene. For example, if a *"TransformRecordComponent"* is specified in the file to keep collisions on, then all instances within the scene will do just that. Finally, each individual *"RecordComponent"* can have a unique *"Playback Ignore"* setup. This will override both the default and the file based approach, allowing for a much more granular approach. For example, this may be useful for when using the same type of *"RecordComponents"* within the scene, but wanting to record using one set, and then playback using another while still retaining user input during playback.

A good example for using the "*Playback Ignore*" system could be when recording a participants' virtual character. During recording the character would take inputs from the player (such as from a keyboard), and the inputs would adjust the characters movement. To prevent accidental inputs by the analyst during playback, the input script can be disabled through the "*Playback Ignore*" settings. This ensures that only the recorded data would be used throughout playback, without accidental secondary input.

5.2.5 Observer Design Pattern

PlayRecorder implements a variation on the observer pattern (Wikipedia 2021g), to both reduce impact on the underlying code of existing systems, and to improve automation of logic recording. It includes a number of built-in events, otherwise known as Actions in C#, that allow for quickly expanding and hooking into core functions of the recording, playback, and components of the system. By including these events, it allows for researchers and developers to create their custom scripts, and then simply retrieve different pieces of data when required. Removing the need to explicitly reimplement update loops reduces developer complexity, as there is no need to repeatedly implement time critical functions, while also improving overall system stability. This alleviates several possible performance issues, meaning that scripts should only need to receive the desired data as it changes, rather than having to request it.

Each part of the system can easily be reached, where required, and is built to be automatically aware of required elements. For example, the recording manager understands every *"RecordComponent"* within the scene, and thus assigns itself as the manager for each component. Each *"RecordComponent"* can then request information from the manager, while the manager sends updates and action calls to the components.

This is used to ensure that items being recorded are uniquely identified within the scene, while allowing for the analyst to modify descriptors of different items. The developer can then link directly to the actions on the *"RecordComponent"* instead of the manager. As each *"RecordComponent"* is designed to be attached to individual parts of logic within the scene, this makes greater sense rather than making each script that wishes to interact with recording simply "phone home" to the recording manager. The playback manager works in a similar way, where *"RecordComponents"* are either automatically or manually assigned within the playback manager, however, do not include a reference to the manager. This means that the manager will understand the components, but the components themselves do not understand the manager. Through this it facilitates the need to rely on events and controlled updates, without having to directly reference the playback manager code.

The individual *"RecordComponents"* within a scene control unique *"RecordItem"* objects. These are updated and stored with a *"RecordComponent"* but are entirely separate to the underlying working of them. This makes the *"RecordItems"* easy to move and transfer between objects, especially when it comes to serialisation. As they are controlled by the *"RecordComponents"*, rather than any of the managers, this reduces overall manager complexity making it easier to understand logic flows. Managers send updates to the *"RecordComponents"* and each of the *"RecordComponents"* independently update their own *"RecordItems"*.

5.2.6 Messages and Statistics

"Messages" are features of the system, as described during the Recorded Data section (Page 114), that allow for simple text based information to be stored during recording. These were expanded to include a selection of statistical messages that allow for the joint recording of numerical, Boolean, or secondary string information. All messages are stored as an array within a *"RecordItem"*, allowing for multiple varying messages to be added to it dynamically as they are recorded. By adding these pieces of information to the recording, the information extracted can be easily streamlined in a number of ways.

Firstly, if simply making use of pure messages, without statistics, then the recorded file will have accessible "markers" throughout it. These can be easily seen and understood using the Timeline tool, explained in full in 5.2.7 Timeline, allowing for the analyst to jump to the relevant parts of files quickly and effectively. Similar to that of markers within video editing applications, these simply store an array of ticks signifying every instance of the message during the recording, and their text based descriptor. This allows for custom logic within the simulation to store key events or actions, that can be used for analysis in future.

Next to this, by utilising the statistics extensions to the messages, the recording can quickly and effectively store statistical information throughout the recording. In and of itself, this is not a particularly useful or beneficial feature when compared to simply recording through a CSV file, and to a degree is incredibly similar in resulting effect. However, when combined with the fact that statistics can be recorded over time, and in turn updated through their playback, it becomes a more powerful tool. For example, if the average velocity of an object is recorded every time a participant releases it, there may be multiple instances of said statistic recorded throughout. Only one of these instances may be the desired value, such as the last instance before a success event. By referencing the success message coupled with the release message, the information can be quickly obtained. This level of expansion means that you can record the entire set of objects and their respective changes, specific events that occur during a study, and varying statistical values all from with the same toolset, and then quickly navigate through all the data and extract it.

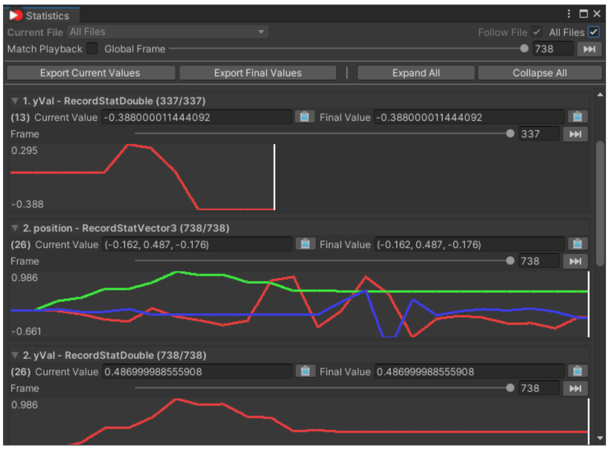
Coupled with the statistics and base level messages, extra analytics can be obtained from recordings. As the system is designed to store all the movements and changes within the simulation, it can quickly and effectively recreate information. As an example, we will use the above example where the velocity of an object is recorded at the point of release. We instead want to find out the average velocity during the time in which they were holding the object. During recording we added messages into the system that were fired when participants performed any interaction with an object, such as grabbing, pressing, or releasing. These are admittedly not critical to our solution as we could always infer when the object started moving through participant means. To go alongside of this, we measure the differences in position of the object between playback ticks, instead of Unity's built in update rate, allowing for consistent data collection. Once we can be assured that the object is being grasped, we simply start recording the velocity for each of these ticks into an array. When the object is released, we stop recording the values, then simply average the array of stored values. These final values can then be stored into a CSV, or similar, file to be then brought into an analysis toolkit. While this may not produce as clean of a method as the integrated system, the flexibility to record such an action is entirely possible, compared to a video or static CSV file.

Due to the nature of how the messages are recorded and stored, both duplicate messages and a lack of expected messages may occur during recording. Messages are only stored when they have been fired from within the code, or by some pre-defined methods. This can lead to instances of having different sets of messages between files, where certain messages may or may not be present. By design, if multiple *"RecordItem"* objects are told to store messages, they will check to ensure there are no duplicate messages within themselves, but not throughout the recording as a whole. If there are a significant number of messages throughout the recording, then this can introduce complexity when analysing data.

5.2.6.1 Statistics Window

The statistics window is a tool available outside of playback, that allows for file's statistical values to be viewed over time. This tool includes a built-in CSV exporter, allowing for fast data extraction from recordings where statistics were recorded. As statistics are recorded with a time stamp, the values can be specified based upon their time, rather than just the final value.

This tool will automatically understand and present statistics from within the currently loaded files. If there are no statistics recorded, then the tool will state as such. These statistics must be recorded at time of creation, adding them into the file post recording is not currently possible.



1. yVal - RecordStatDouble (337/337)
(13) Current Value -0.388000011444092 Final Value -0.388000011444092
Frame 337

2. position - RecordStatVector3 (738/738)
(26) Current Value (-0.162, 0.487, -0.176) Final Value (-0.162, 0.487, -0.176)
Frame 738

2. yVal - RecordStatDouble (738/738)
(26) Current Value 0.486999988555908 Final Value 0.486999988555908
Frame 738

Figure 91 The statistics window showing three different statistics and their graph visualisations. The visualisations are dynamically reactive to the data type and length of the recordings visible.

To help improve understanding of values and the data, simple graph based visualisations are included within the tool. With these visualisations, the researcher can confirm and verify that the statistics captured are the correct values before they were to extract and export them into a different application. These are fully mouse interactable, allowing for the analyst to quickly jump to specific values within the file. All these visualisations are scaled to their maximum and minimum values and support multiple axis values such as 3-axis vectors. Adjusting the values of these visualisations changes the "current" value of the statistic. The analyst can then export all the "current" values, or the "final" values into a CSV file. Including the option of a "current" and "final" values means that the statistics can be customised to obtain values that may be specific to events. For example, if the same statistic was used for measuring the direction of a hand when it grabs an object, but there are ten objects in the scene. The analyst could see and log the directions of a specific object by simply scrubbing the statistics' current value to match the correct timeframe of when it occurred.

5.2.7 Timeline

The timeline is an included tool used for simple analysis during playback. This tool will quickly and effectively visualise where messages were recorded during every single currently loaded file and is thus heavily reliant on the usage of messages during recording. By directly tying into the ability to load multiple files into the playback manager, it also allows the analyst to quickly jump between files and scrub to specific time positions within files.

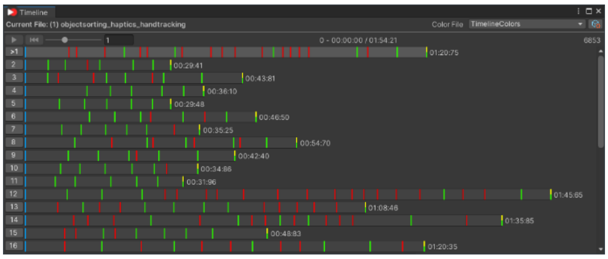

Current File: (1) objectsorting_haptics_handtracking
Color File TimelineColors

Figure 92 The timeline window.

The above figure shows the visual output of the timeline, with different messages recorded in the files with varying colour coding. Each message is represented as a bar, and each file is represented as an individual track. Every visual segment of the timeline is customisable, be it the colour and transparency of individual messages, or the colour and sizing of different segments of the interface. By introducing these customisations to the timeline, the analyst can quickly filter out specific messages and focus in on particular parts of their recordings. For example, within the block stacking portion of the FMS study, we were able to colour code varying messages to help speed up navigating through the files on the timeline. As the task had two different sections, with different success and failure states, we could quickly filter the information by using one set of colours for the stacking and one for the removal stages.

The timeline operates through the usage of the cache created by the playback manager. This means that the messages have been "flattened", where all messages for a recording are grouped into one set, irrespective of the different `"RecordItems"` they may have been attached to. Due to this the timeline will not directly report every single message, with this partly being done to reduce visual clutter. Duplicate messages will be concatenated into a single visual bar. If multiple different messages are present for the same time frame, then message bars will be vertically split. By combining the colour coding of messages and ability to scrub between times and files it can significantly speed up the process of data understanding. This can be especially useful for finding and understanding anomalies and edge cases within data sets.

5.2.8 System Expansion

By design, PlayRecorder is meant to be expanded and modified to meet the custom needs of any study. This meant including a set of default functions that are designed to be overridden and replaced through inheritance. As an example of how these parts of the system can be modified, we will look at the conversion of the default `"RecordComponent"` into the `"TransformRecordComponent"` (TRC), the extension that allows for the recording of transformation data (position, rotation, and scale) of an object.

The TRC overrides the default recording logic within both the recording thread and Unity main thread. Each of the base level frames are extended into `"TransformFrames"` to store a position vector, rotation quaternion, and scale vector variable. At its core, the TRC uses a component cache to store the latest difference observed for the transform component of the object. If any of the three core properties of the transform have change then it will report it to the cache. These changes are monitored during the main Unity thread, and then if the cache reports it has been changed it will be recorded as a new frame during the recording thread function. Once changes have been recorded, the cache maintains the current values and will then continue to test against these for future changes. When recording is completed, the information is collated into a standard `"RecordItem"`, no extension needed as we're performing the majority of the logic within the TRC and then storing the data within the custom `"TransformFrames"`.

Playback functions of the TRC are also overridden, however, it does not need to override the threaded playback function. Playback ignores are also set by the script, where the TRC is set to disable all physics, and collisions as these could negatively impact playback consistency. During the playback thread update, the TRC updates the current tick as per normal and sets any changed `"RecordParts"` of the `"RecordItem"` in the updated parts array just like any other `"RecordComponent"`. Once the main thread playback update is called this is then iterated over and applies each of the values of the current `"TransformFrame"` to the transform of the object.

This example of expanding the system is achieved within under 200 lines of code, requires no modification to the managers or underlying data saving, and works completely with all existing parts of the system. Just as the base `"RecordComponent"` can be overridden, as can the TRC and any other custom implementation. All recorded data was correctly recorded and loaded by simply expanding upon the base classes as needed.

5.3 Implementation Case Study

In this section we will cover the use case PlayRecorder was originally built for, versus a prior project. We will explain how it was set up, what was recorded, and the benefits it provided within the Fine Motor Skills (FMS) project. Comparisons will be made against the 3D Button Reactions (3DBR) study, which utilised conventional screen recording methods, and JSON recording for certain statistics. These are the two studies found in chapters 3 and 4 for the 3DBR and FMS studies respectively.

5.3.1 The Studies

The FMS project consisted of a participant either stacking three size decreasing blocks into a specified area or sorting three visually different blocks into containers. Participants would be interacting using either optical hand tracking or VR controllers that provided finger pose tracking (Valve Index controllers). Users would complete each task with haptics being applied to half of their trials. This resulted in eight individual task combinations (two tasks by two input methods by two haptic conditions), which were repeated twice and thus resulting in sixteen total task completions. The study operated within one single scene, and simply adjusted the scene based upon the individual task requirements, updating the component states as needed. Participants would receive a small in-study NASA-TLX style survey after each individual task.

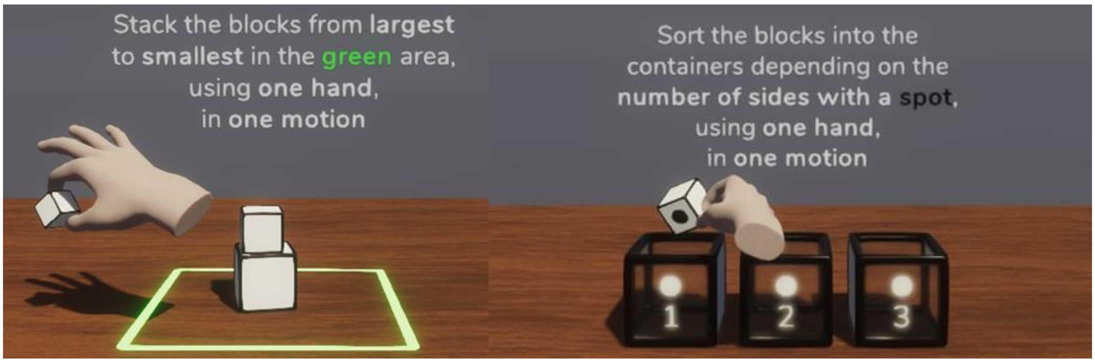


Figure 93 Two images showing the tasks of the FMS project. The block stacking task on the left, and the object sorting on the right.

3DBR consisted of participants working through three sets of six 3D buttons, with varying types and levels of reactivity, to turn a light on and off. The three different sets of reaction consisted of translation, colour, and physical shape changes. Each set would have a low, medium, and highly reactive set of buttons, with a mid-air ultrasound haptic and non-haptic counterpart for each, resulting in six buttons. Participants would only be interacting with the study using optical hand tracking. After each set of buttons, the participant would rate their opinions of the buttons and the haptic conditions.

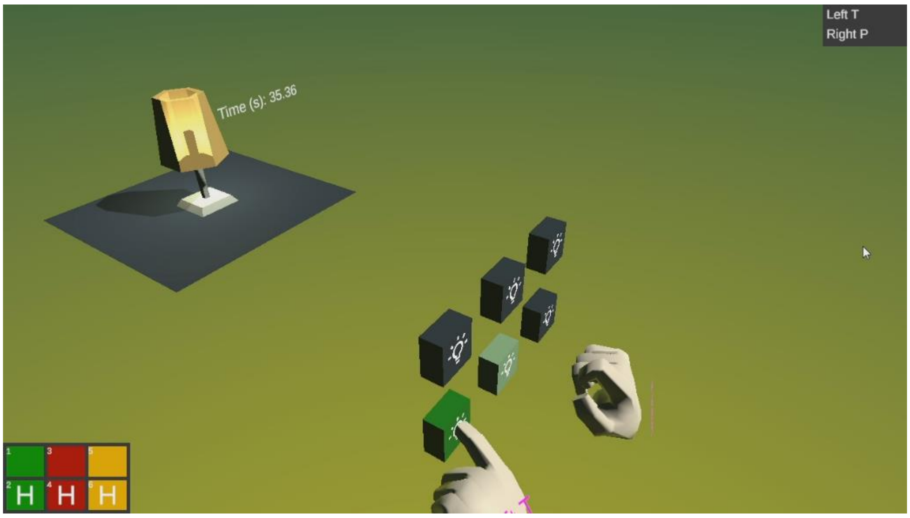


Figure 94 The 3DBR study. In this instance the participant is interacting with colour reactive buttons. Information about the hand states were present in the top right and on the backs of the hands. The bottom left of the screen shows which buttons were the low, middle, and highly reactive with the "H" denoting which buttons had haptics being applied.

5.3.2 Recordings

PlayRecorder was used through the FMS project to record all participant movements and actions, study logic and completion, and their in-study survey answers. We recorded participant movements and actions to understand the varying effects of using different input methods and the overall effect of haptics. This was done at 60 frames per second, while the study was undertaken using the Valve Index which reported a visual frame rate of 144 frames per second. As we were recording relatively slow interactions, we had no need to use a higher frame rate, as the interactions would still be recorded in their entirety. This frame rate struck a good balance between information collected, as well as the impact on the system when recording, which was especially critical as we were running at such a high frame rate. If we had been recording a user performing high speed movements, such as hitting a virtual tennis ball with a racket, then we would have wanted to use a higher frame rate to ensure we recorded all the possible information changes. Recordings were undertaken within a fully built application, with playback occurring in the Unity editor.

During the 3DBR study, each participant was recorded through OBS, from a fixed offset viewpoint. Recordings were performed at 30 frames per second, with the simulation running through the HTC Vive with a visual frame rate of 90 frames per second. Visual information was displayed on the screen showing hand poses and information about the hands in operation. Information about the participant was recorded in custom classes and stored into JSON files, encompassing times throughout the study, survey answers, and the positions of buttons.

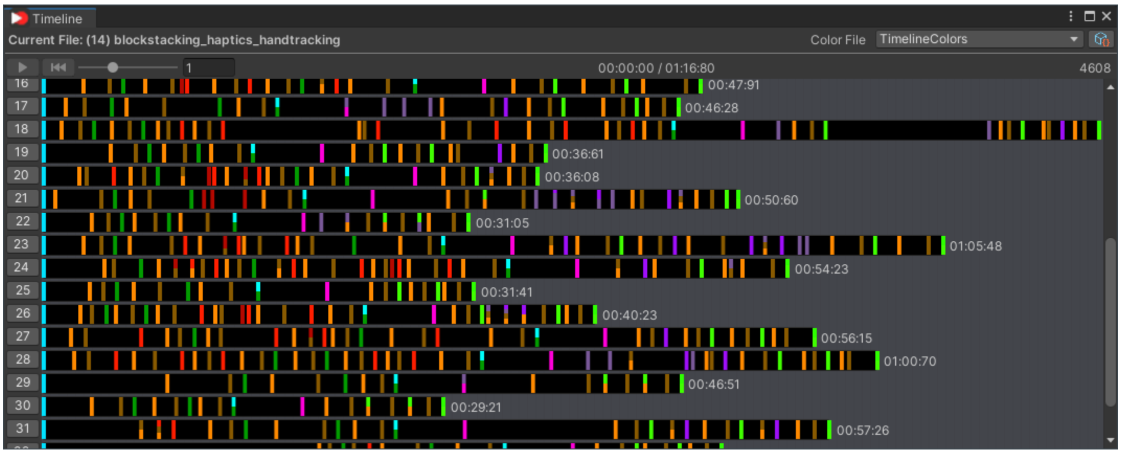
All logic for the FMS user study was performed within a single, dynamically changing Unity scene to reduce overall system complexity. Tasks were completed in an entirely randomised order. Each recording was taken when the participant started the individual tasks of the study, with interaction recording being paused once they had completed a task, and then subsequently stopped when they had completed the post-task survey. Depending on the setup of the task, the system would record either the participants hands, or the controllers. This information was decided by the study logic, which controlled when recordings would begin. Every recording was automatically started and stopped by the simulation to ensure consistency between participants.

The 3DBR study operated within one overarching scene, which controlled the hand logic, and three individual scenes, which handled the sets of buttons. Tasks were completed in a randomised order, with buttons being randomly placed by reactivity level, and then secondly by their haptic condition. Visual recordings were manually initiated by the study administrator at the beginning of the entire study and stopped once all tasks had been completed. JSON files were automatically saved once the participant had completed the final in-simulation survey.

Custom *"RecordComponents"* were developed for the FMS study, which allowed for the recording of the study survey questions, without having to record object interaction logic. These components simply stored integer values into their *"RecordItem"* as custom variables, extending off of the base *"RecordItem"* through the usage of inheritance, without storing any frames. This meant creating a small set of custom scripts that inherited the default *"RecordItem"* logic and variables but included extra custom ones for task values. One was created for the block stacking task which included times for the stacking and removal stages of the task, with another being created for the object sorting task which included the overall time for the task. Secondly, there were custom components developed to record the information from the Ultraleap hand tracking input and SteamVR skeletal input systems. These allowed for the digital representations of the hands, reported through either the camera optics or the varying sensors on the controller, to be recorded.

5.3.3 Data and File Sizes

PlayRecorder demonstrated significant reductions in storage requirements for the study. A total of 280 recordings were made for the FMS study, resulting in 3 hours and 58 minutes of recorded data. This data had a total size of 1.48GB, which in turn gives a value of 0.78 megabits per second. As mentioned, all recordings were performed at a 60 frames per second speed and are not restricted to a resolution value as they are rendered through the Unity engine. Average 1920 by 1080 resolution 60 frames per second video formats range from around 10-50 megabits per second, making it significantly smaller in video size. Comparatively, the 3DBR study had a total of 23 recordings, resulting in 2 hours and 40 minutes of recorded data. The recorded videos had a total size of 3.07GB and each JSON file was around 1-2KB, these resulted in a value of 2.56 megabits per second. Overall data rates were 3.3x for the 3DBR study compared to FMS. When adjusted to match frame rates (30 frames per second to 60 frames per second) that value is doubled to 6.6x, and when adjusted for resolution (720p to 1080p, a pixel change of 2.25x) that total becomes 14.85x, resulting in an estimated value of 33.5 megabits per second for comparable visual results. While this value cannot be truly accurate due to differences in compression techniques and video formats, it gives a quick estimate into possible differences when working with competing resolutions and frame rates.



Current File: (14) blockstacking_haptics_handtracking

Color File TimelineColors

Figure 95 The timeline showing the different colour coded messages from the FMS block stacking task. Each of the messages could be coded with different colours, or hidden, within the timeline. In this instance, the stacking segment of the task started and ended with the cyan colours and had dark greens and reds for success and failure states. The removal segment of the task had pink and purple colouring for success and failures. All the grabs and releases were presented with orange and brown respectively. This allowed us to quickly see when and where different events occurred throughout all of the participants data, without having to explicitly watch all of the files. Each step of this setup, having multiple files loaded, being able to see the messages and events, and being able to customise each message, made it significantly faster to analyse and see trends throughout the files compared to the 3DBR study.

Data was loaded into Unity with each study condition loaded separately, with 35 recordings per condition. To enable faster changing between conditions, we saved each set of files for each condition into a playlist. A secondary study specific set of tools were developed to work alongside PlayRecorder, allowing us to collect statistics. While we could have included statistics storage into the study before recording the data, we were unfortunately time restricted due to COVID-19 situation at the time of the study. As there was no formal statistic collection feature at the time of study participation, or definitive way of collecting the quantitative data, we chose to collect most of our statistics post-participation. The level of accuracy and overall quality of the simulation recreation that PlayRecorder provided made this an entirely viable, and easy to work with, solution. These tools directly recorded and analysed movement and interaction data from the objects such as object grab time, release times, and direction of grab. Each of these statistics were automatically recorded into a cache, which was then stored into a CSV file for analysis within Python. Visual cues about participant motions and completion anomalies were documented, allowing us to help work with possible outliers. This was only possible through the usage of PlayRecorder as it allowed us to freely move around the hand and selectively hide objects within the scene to improve visibility.

3DBR study videos were individually loaded into a standard video player, in this case VLC. All visual data from the 3DBR study was split by participant only and was not automated, meaning timings from video recordings were not entirely accurate and included extra unnecessary elements such as interactions during survey questions. Information was observed from a single fixed viewpoint, with hand state data being presented onto the screen. While this information was easy to obtain from the videos it

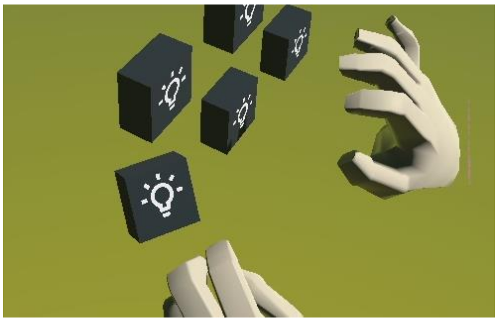


Figure 96 An example of hand information being cut off by the viewpoint in the 3DBR study. There was no way to understand the full pose of the left hand for this button interaction, and in turn made it challenging to obtain consistent data.

was however an entirely manual process. It was impossible to obtain further consistent information from the videos, and the lack of segmentation between tasks in the video made it cumbersome to swap between participants and files. The fixed viewpoint of the recordings made it challenging to always see the hand based information, with multiple instances of hands being slightly out of frame. The associative JSON files were used to analyse timings and study questions, as well as understand where the buttons were within the scene.

Overall, when compared to the 3DBR data collection method, PlayRecorder provided significant and repeated benefits on a multitude of fronts. PlayRecorder considerably reduced the file sizes we were working with, at a rate of 70%, which vastly improved the efficiency when analysing data collaboratively. If we had only recorded statistical information, such as grab times or velocities, then it would be impossible to understand anomalies or possible erroneous data, thus making it harder to understand outliers. For example, we had a participant that had a significant number of repeated error states in one of their object sorting recordings. Through viewing the timeline, we were able to navigate to the instance and then able to see exactly what had happened and why. It happened to be that they were repeatedly covering a piece of the visual information with their hand, thus causing an incorrect failure state as they sorted it into the wrong container. Without being able to directly see what happened, the error counts may simply have been ignored or removed during analysis due to being an outlier. The overall level of data detail and flexibility was magnitudes better and more efficient when compared to the 3DBR study, allowing us to produce more accurate and factual results throughout our analysis. We were able to directly analyse visual information from any angle, at any time, with any combination of objects being visible or active, something that would have been impossible with video. While it was initially more cumbersome to work with PlayRecorder during implementation and analysis instead of pure video files, requiring the usage of the Unity Editor, specified implementation of individual `"RecordComponents"`, and having the full project downloaded, every recorded file after that fact was smaller in size and provided greater detail. Introducing extra data into our setup and automating it removed possible erroneous data collection or accidents during the study, which helped streamline the overall study delivery process. Combined with a relatively easy recording implementation, the usage of playlists, messages, and the timeline tool, PlayRecorder made it faster to navigate and analyse files at every stage.

5.4 Conclusion

Within this chapter we have presented our flexible, customisable, data collection and playback toolkit for user studies within the Unity engine. It has shown positive merits compared to conventional video-based screen recordings, requiring less processing power and storage when recording, while providing numerous options during playback such as adjusting viewpoints and interchangeably hiding recorded elements. There are several methods for recording statistical information, including methods for recording statistics over time as well as instantly extracting CSV data. This allows us to combine the benefits of both data types into one system and one single recorded file.

We fulfilled the requirements we set out to achieve, while introducing extra tools during development that help expand the value beyond our initial scope. The tool allows for fully customisable simulation recreations, recording and playing parts interchangeably as required. It is easy to implement data expansions, producing custom elements and logic for the system without having to modify any underlying systems or existing logic. The tool is performant, working through multi-threaded code, and produces lightweight files by taking a change-based approach to recording. We developed features that are easy to understand, use, and re-implementable features throughout the tool, helping to speed up recording, playback, and analysis of data for researchers. All of these features were consolidated into a single toolset that works consistently throughout developing and built user studies, with no degradation in features.

The tool is available as a free and open-source package at https://github.com/ultraleap/PlayRecorder and can be integrated into any current Unity project. Development and feature changes will continue into the future, with the intention of using the tool throughout future research projects.

6 Discussion and Conclusion

Commercial level, real-time hand tracking has significantly improved over the years, with large swathes of research now pivoting to interactions over implementation. The research within this thesis has focused around this, exploring three topics all under the umbrella of understanding the effects of hand tracking on interactions within VR, and how it could be improved. Crucially, the main focus has been around the base level of these interactions, where the results and findings can be applied without particularly specific context.

One overarching theme that has presented itself throughout the interaction studies is that of "less is more". Several of the implementations, especially in regards to haptic feedback, were trying to provide too much information to the user. The process of providing information or helping the user throughout their interactions should be done seamlessly. If not, then the overall perception will be more negative than positive.

Creating novel and unique interactions and technologies always lend themselves to being shown off in the most powerful way they can be. To a degree this can help propel the technologies to the forefront of user interest, however it can significantly impact its longevity. Having a technology that is incredibly good at doing a few things brings interest, but can result in a long lag time between new discoveries. This was evident with consumer level hand tracking as found with the original versions of the Leap Motion, where it took years to reach the level of VR interactivity that it currently achieves. While it finally seems to be finding its pockets to grab onto, with headsets starting to include hand tracking in them as standard integrations, it took significant streamlining of the surrounding tools to ensure that it could happen. Early generations showed multiple impressive tech demos, but failed to gain significant traction until they were able to simplify their "magic".

To an extent, this cycle is what happens with most normal technological advances we currently take for granted. Exactly in line with this, in my personal opinion, is the current state of mid-air ultrasound haptics. It provides something that is hard to fundamentally understand until directly perceived, but has been oversold in certain areas for a number of years. The promise of stopping a user's skin dead in its tracks is not feasible, and is unlikely to ever be realised due to the required amount of physical pressure. It is not to say the technology is without merit or purpose. Many of the haptic feedback effects presented in this thesis have revolved around continuous application of haptics, providing constant information to the user. A gentler approach with greater nuance should provide better results than blasting the users hand off with pure ultrasound. It may result in weaker preconceptions surrounding the technology but could result in greater payoffs as it exceeds the expectation of the user.

6.1 Hand Tracking in VR

Hand tracking has shown significant promise, with more natural and intuitive ways of interacting as outlined in our direct comparison study. The strong prevailing counter opinion is one of a general lack of accuracy and interaction speed. Developers that have historically tried to implement hand tracking into every possible area of their system have often tried just performing one to one conversions. This means attempting to replace their buttons and clicks with obscure hand gestures and movements to perform old actions. While this may have the expected effect of still being able to interact with the system, but the resulting experience will be poor. This is far from the ideal or expected behaviour that original creators would want, however it is a realistic turn of events. Multiple large VR titles have been ported across from 2D screens with no significant changes beyond adding controller support to the original mechanics. They meet the requirements of the game being playable but do nothing in regards to providing a worthwhile VR interaction experience.

Experiences that have flourished in VR have usually included some set of well defined, seamless, helper functions that provide meaningful improvements to each interaction. This is the same when it comes to hand tracking. Improving the method of interaction, through the use of novel physics calculations, has yielded better result than trying to emulate a binary response a controller may provide. There are indeed many more degrees of freedom available to the interaction which does increase the complexity for processing and possible success or failure events, but results in a more natural experience. Fine tuning those variables will be where the true success will be found, ensuring that objects naturally release when intended.

In its current state, I feel as though hand tracking is starting to reach the convergence point. Developments have made it significantly more accurate, provided greater stability, and enabled real-time interactions within VR. The prior issues of accuracy and stability are mostly solved, with the interaction developments being left. While I don't feel as though hand tracking will ever fully replace dedicated controllers, it can easily be a supplementary technology for the majority of contexts. Similar in vain as a steering wheel is used within driving simulators, controllers may soon be shifted towards the realm of specialised interactions, with hand tracking taking the majority of the standard interactions. This would enable systems and simulations to be controller free for most people, without having to fully sacrifice the controller where needed. In some training sectors this already partly occurred with 3DOF headsets rarely including controllers, the inclusion of hand tracking to these types of systems would retain that portability while significantly heightening immersion and learning effects.

As more devices include different novel sensors, the possibilities for their inclusion could help improve interactions. Eye tracking could play a big part in the improvement as visual context and intent could play a large role in understanding what the user is wanting to do. Studies have found that it's definitely viable for performing gesture based work (Pfeuffer et al. 2017), however the field of combined direct 3D manipulation is less studied. Understanding user intent would be incredibly potent as the system could have a grasp on the user's trajectory and desired actions before they are performed. As the available processing power is increased, the possibilities for implementing this, especially within self-contained devices, increases once again.

6.2 Mid-air Haptics

Mid-air haptics are still one of the youngest technologies within the haptic space. As this research has progressed, developments outside of this project have helped focus the trajectory and method of implementations available. The overall strength of the haptic feedback, coupled with the rendering methods were improved, allowing for heightened perception and greater control of feedback. Differences can be observed between the studies, where participants had more divisive and strong opinions within the second study compared to the first. This can be attributed to the difference in study design, however the improvements in methods of application cannot be understated.

While the implementation of haptic feedback to the different buttons in the first study resulted in a generally positive outcome for the user, there was less of a distinct overall sentiment towards them than we expected. This was something also noted throughout the object interaction study, even when the implementation and context was significantly different. Many pieces of work sing the praises of haptic feedback as the grand solution for improving an experience. This is often the case with new technologies, where every possible application will be tried at full power as the overall benefits and negatives may not yet be fully realised. While haptics can improve the sense of perception to a degree, an overall light handed approach is needed to ensure effective implementation. Simply blasting the user with haptic feedback, especially novel types, may result in an initial wow factor, but quickly transitions into resentment, annoyance, and even at times mild pain. Similar findings were reported by (Rutten and Geerts 2020), where they discovered no significant differences once the initial novelty factor was removed.

There is still plenty of work to be done on mid-air haptics, the field is still in its fledgling stage where optimal use cases, let alone rendering methods, are still to be discovered. The COVID-19 pandemic has provided a push into a possible direction of out of home interactions, however this is still unproven ground. Within VR there are plenty of possible use cases, however they still require the user to be sat down interacting with the device in-front of them. Until there is an effective solution for mid-air haptics that can be attached to the user, the realistic application of the technology within VR and AR will be limited.

6.3 3D Buttons

Button reactions are a key factor for any user interface, even if they're only a small part of the whole system. They directly affect and present state information to the user at every step of their journey, with the resulting effects changing their perceived experience. As shown in our research, the types of effect can be highly opinionated.

Designers like to provide experiences within their interfaces that delight the user, usually by including special extra effects and animations. To a degree this can be beneficial, however our user study showed that as the type of reaction significantly increased in complexity, the overall perception by the user generally degraded. Many opinions were stating that while they found the differences and changes of the interface entertaining, they were overall relatively distracting. Ensuring the interface is both effective, as well as seamless in its working should be paramount. The best types of interfaces are the ones that can provide everything the user needs and wants with minimal effort on their part, enabling the user to perform at their highest. There is a reason as to why minimalist interfaces are so popular at the moment and the research here helps back that up.

Just as with websites across the internet, there are going to be good and bad implementations of buttons within VR. While it may seem like a small part of the overarching experience, ensuring a level of consistent quality across the interface should be paramount. As shown with other research studies, the more logical the effect of reaction, the better the button performs. The inclusion of fundamentally different techniques, such as real-time deformation, are tested on too small a sample size and time period within this work to fully understand their differences.

6.4 3D Object Interactions

3D object interactions have historically been performed entirely through proxy materials, be it a tool or set of widgets on the screen for the user. As VR has become more common, that sentiment drastically changed, with users able to use 6DOF controllers to interact with the environment. That is starting to shift once again with the increasing popularisation of hand tracking, where users are able to simply grab and manipulate objects with their hands.

Recent developments, as shown within this work, have allowed for hand-based interactions to shift from gesture derived proxy interaction, into direct finger recognised manipulations. The accuracy and latency currently available has helped the technology reach a stage where the comparison against controllers is no longer the black and white comparison of one being distinctly better than the other. Differences between technologies is far more nuanced with the gaps being reduced to a point where hand tracking is significantly more viable for many interactions than prior thought. The research in this thesis is testament to that where each task was successfully completed by a user with minimal stability issues, which improved between projects. As the two projects used slightly differing technology stacks, the benefits between them was apparent, with heightened accuracy and fewer issues between.

The initial battles of proving hand tracking as a viable input method are starting to fade. Many device ecosystems are starting to adopt it, especially within the self-contained device market. Large amounts of effort will shift from the pure implementation of the technology, which will converge across sectors and vendors, towards that of how to effectively work with it in varying digital environments. Research shows that there are promising tools and methods already available, but still require further refinement. Unlike gesture interfaces that have predated these interactions, the barriers to entry should be considerably lower, and in turn provide better experiences to the user.

6.5 Real-time Data Collection and Analysis

Data collection has improved as time has gone on, however still relies on mostly the same basic approach for many interaction studies. The method of visually recording the interactions provides no significant information about the statistical positionings or other key information about the system. On the flip side, many of the studies that do record statistical information tend to still lack any significantly easy was of analysing their visual outputs, with issues such as occlusion being prevalent, especially with high field of view HMDs.

PlayRecorder has shown promise in delivering these combined benefits of both visual and statistical recording, without significantly impacting the systems. The reality for many research studies is that the implementation or idea of data collection is one of an afterthought. The developed toolkit helps in this approach by not requiring significant overheads or extra processing power from the system to still effectively use and implement it. There are areas of the implementation that could definitely be streamlined, such as the requirement of adding scripts to the individual objects that need to be recorded, but the payoff for the amount of required work to implement is significantly larger than with video recording.

A key set of interesting points noticed throughout the literature review was that many studies lack significant quantitative analysis for areas that would significantly benefit from it, such as device or interaction metrics. Hopefully through the developed toolkit we can record this type of data for the researcher, and provide easy extraction of the information for further analysis.

6.6 Closing Remarks

Within this thesis I have presented the development, testing, and analysis of two VR interaction related user studies. Coupled with this, the creation and development of a real-time user interaction recording toolkit is covered. These have all been undertaken with the key focus on the effectiveness of hand tracking within VR, critically comparing it to different input modalities and developing tools to understand the effectiveness. The novel technology of mid-air ultrasound haptic feedback was applied through the studies to research the effects of this feedback and whether it can be utilised as an effective and reliable method for applying haptics.

The first user study covered that of button reactions for 3D user interfaces within VR. Three different modalities of visual reactions were tested, these being colour, movement, and real-time deformation. Each of these buttons represented different types of feedback found within different conventional interfaces and 3D visual techniques. Different levels of these reactions were compared, allowing for greater understanding of the individual nuances of each type of button. Secondary effects of mid-air haptic feedback was applied to see if could be used to improve the reactions presented to the user. The study was heavily focused towards user opinions and overall impact on their feelings, with in simulation surveys being used coupled with post study surveys. Within this study it was found that the more novel a reaction the more confusing they were, with common and physically accurate reactions being easily understood and favoured.

The second study covered the topic of 3D object interaction, where I tasked users with two different fine motor skills tasks. These tasks were designed to assess and analyse varying performance metrics of the user explicitly and intrinsically. They were based upon commonly used tasks within fine motor development study of children. Each user completed the two tasks using both hand tracking and controller input methods. Both sets of inputs were compared against each other, as well as with the application of haptic feedback. In the case of hand tracking this was provided through the use of mid-air ultrasound, and for controllers this was produced through their internal linear resonant actuators. Study participants interactions were recorded using a custom recording tool that allowed us to reconstruct and play back their trials in full 3D, allowing for heightened analytical options. Individual metrics and statistics showed significant differences between different input and haptic conditions. Hand tracking showed either better or comparable performance with controllers in many categories, with grab times being especially faster. Controllers were shown to be more effective at releasing objects as well as having heightened accuracy of interaction.

Both of these studies fuelled the development of a real-time user study recording toolkit called PlayRecorder. This toolkit allows for researchers to record their user studies and then fully recreate each trial within the Unity game engine with no additional changes. Unlike conventional video recordings the files are recorded changes and states of each object instead of pixel information, significantly reducing file sizes. A suite of surrounding tools were created to allow for researchers to quickly and effectively analyse key metrics of their studies, such as custom messages and statistic recording. As the recorded data is the pure information of each object, it allows for the researcher to easily precure extra statistical data that may have been missed during the initial recording. This toolkit was subsequently open-sourced for other researchers to use within their work, designed to be implemented with minimal impact on the researchers' existing applications.

6.7 Future Work

The research as a whole has been focused on the interfaces, object interactions, and improvement of analysis of real-time hand tracking, with substantial effort into the implementation and union of mid-air ultrasound haptic feedback with said tracking. While the work has shown significant findings in contrast to different interface designs, input modalities, haptic implementations, and analytical tooling, there is still further work that can be done in these fields.

6.7.1 VR Button Reactions

While we have tested our current mechanics, further work is required into prolonged testing and the interfaces. A number of these techniques have initially proved challenging for users to understand but may be favourably adopted with further testing. This would help reduce the learning effect associated with the buttons, and may change overall sentiments.

Although we have presented the findings into unique button modality changes, further research into combinations of these is required. Interchanging the different button types may result in better outputs, as this could allow for different modalities to provide different reactions at different stages. For example, buttons could have a movement method of transition as they're being pressed, followed by a colour output to signify completion of action. In a similar vein, the addition of sound could help reinforce the effectiveness of both the button modalities as well as the application of haptic effects.

As observed within other guidelines for designs, our recommendations only apply to the buttons without significant contextual application. Introducing secondary effects to the system could help reinforce the overall views and opinions of the buttons, while also exposing secondary unforeseen effects. We expect designers to implement these guidelines into varying larger systems, work which is beyond the current scope of research.

6.7.2 VR Object Interactions

The results of the object interaction study have presented us with a set of interesting findings; however, further work could be undertaken in a few key areas to improve study findings or explore different interaction tasks, and thus resulting interaction methods.

Adjustments to haptics are required to fully understand their effect on the object interaction, and overall performance on participants. The current implementation was made to provide as fair of a comparison as possible between input devices. While the implementation of mid-air haptics does provide a successful hit rate (haptics being applied to the hand instead of being missed), the usage of it may have been too broad. Our desire to provide extra information to the participants may have resulted in an overload of sensory information and may need to be tweaked to reduce the onset of this. This means there are further possible findings and benefits that may be present within individual input modalities once haptics have been adjusted.

Changes to the overall visual representation of hand-object interaction could be made to provide a more realistic resulting effect. This would mean implementing virtual hands that do not penetrate the objects, and realistically wrap around objects during a grasp. Participants may start adjusting their physical hand poses to match these visual changes, as they would have already passed the required threshold for interaction. Doing so could result in significant differences to the participants hand poses, while also resulting in changes to perceived haptics. As ultrasound haptics are targeted on the palm of the hand, the participant would be more likely to receive them if they were to match their hand pose to the virtual when grasping a relatively large object. Small objects would receive less of a secondary benefit from this as the participant would still be naturally closing their hand around the object.

The participants were shown to repeatedly struggle when releasing objects. Further adjustments and development into how objects are released could cause significant overall improvements to the interactions when utilising hand tracked input methods. As the study did not directly modify the provided toolkits underlying logic for interaction methods, this would be a considerable undertaking.

Several participants were shown to adversely avoid the overall requirements and desired intentions of the object sorting task, while still being able to complete the required objective. As there was no visual restriction on the object before they grasped the block, the participants were able to physically manoeuvre themselves around the object to see the information before proceeding. Restricting visual information until the object is grasped would help reduce issues with unintended successes and focus the participant back towards the grasping interaction.

In a similar vein, the block stacking task had a number of participants creating unintended error rates due to their overzealous speed of completion during the dismantling stage. Reducing speed until success checks are completed could be achieved by adjusting visual and interaction options for the objects, such as by reducing visual opacity of the object and preventing any hand-based interactions with the object. Doing this may result in more consistent data during the dismantling stage, as several erroneous events occurred due to unintentional grasping of non-current blocks. This could also result in unintended accuracy negligence by the participant as it might end up removing the effect of knocking the tower when removing the current block.

Statistical analysis may be improved through a few adjustments to the study design. By reducing the number of main effects, we could produce more robust data analysis, which may in turn produce clearer results that would be easier to analyse. This in turn would provide clearer results and more concise discussion and delivery of information. Introducing a time limit to tasks, even if the time limit was sufficiently easy to achieve, should result in less deviation in results, and thus produce data more likely to be normally distributed. It should help amplify pain points in certain effects as this should exacerbate failure rates and similar undesired instances that may be present within the different factors, such as input or haptics. Adding a third repetitions of trials could help iron out any particular learning effects that may be present within the study, while providing an extra 50% increase of data points across the board.

6.7.3 PlayRecorder

PlayRecorder has such a wide level of scope that it almost entirely lends itself to perpetual "feature creep", the process of continually adding little pieces of logic and new features without ever fully releasing or finalising the development. There are however a few specific features that could significantly improve the toolkit, both for its ease of use as well as its effectiveness.

Firstly, the introduction of a standardised caching system for `"RecordComponents"` updates will improve the ease of their usage. The current implementation is done through individually specific setups which are unique to each point at which they are used, even if they are admittedly easy to extend, such as the transform cache used in the `"TransformRecordComponent"`. Collating this into one unified system will improve the ease of implementation, as well as ensuring it follows the rest of the ethos of PlayRecorder as a whole.

Secondly, adding further features to enable the editing and modification of data post-recording will help improve the overall flexibility of the system and all subsequent data. Currently there are a very minimal number of features that allow for recordings to be modified during analysis or playback stages, the most people can do is set the colours and theming in the timeline tool. Features such as trimming or splitting of files could allow greater organisation of data.

Thirdly, modifying the recording process to support chunked, or segmented, recordings could improve recording stability throughout studies. The current method of recording relies on storing every part of the recording into the current memory (RAM) of the system, collating the information at the end, then serialising and storing it into a final file. This method works effectively on systems with large amounts of available RAM, however, could negatively impact the performance of lower end systems. By implementing this we could also help ensure that recordings are still recorded even in scenarios where a simulation may crash, whereas currently it will cause the unfortunate by product of being lost.

Fourth, the inclusion of frame interpolation could help improve the visual output during playback. Currently the system will simply update the information at the specified frame and time, without any understanding of the journey taken to achieve said update. While it may not be feasible to record every single possible update to an objects motion, especially when working at incredibly high frame rates, the usage of interpolation could automatically smoothen out perceived stutters in motion. This would require changes to both the playback of the system, as well as frames of the recording to ensure that data does not interpolate unintentionally.

6.7.4 Future Studies and Questions

The work presented here summaries that of two key studies, with development work undertaken for an additional section. Both of these studies could be expanded in their scope, trial variations, implementation, and overall execution.

6.7.4.1 VR Button Reactions

3D button reactions could be implemented within a realistic context, such as an application or game. This would reduce the instantaneous focus of the user on the interface, and thus present results that are more realistic to those found in production software. It would also help to cement the statements made within this work. It would also be beneficial to test the interface changes over a prolonged scenario of several days or weeks, which is fundamentally more challenging within an academic setting.

Complementary to this, the reaction effects could be implemented into a cohesive interface, where they would be complemented with elements such as menus, toolbars, and panels. This would once again, help to expose issues and reinforce findings from this study.

Another key question could well be available with advancements to processing techniques and technology. The haptic implementation could benefit from a change to the underlying hand palmar surface placement algorithm, which is currently challenging due to the optical hand tracking accuracy. If these issues were overcome, does the overall perceived effectiveness of the haptic feedback change?

While most studies try to limit their variables and scope, this study could be enhanced with the introduction of varying types of modality combinations. This could unlock greater information where certain modalities excel, especially in instances where they are combined with different levels of reactions.

6.7.4.2 VR Object Interactions

While the study covers a comparison between the Valve Index controllers and optical hand tracking, these controllers include features that are less common than other controllers. Including a set of controller inputs that rely purely on controller button presses, rather than pressure sensors would result in an interesting extra input modality to compare against. Similarly, the hand-based logic for grasping objects relies on the use of multiple fingers for grabbing objects. While this is great at replicating a realistic approach to grabbing objects, it's distinctively different to the binary approach of many controllers. Moving towards a heuristics approach that relies solely on the curl of the fingers to report a "grabbed" or not value may result in significantly different results.

Changing the haptic implementation of the study to that of status change information, instead of presence, may result in a significant difference to the results. The change itself would be technically easy to achieve, and may even require less physical accuracy of the mid-air haptics, but should distinctly change every opinion of the haptic feedback.

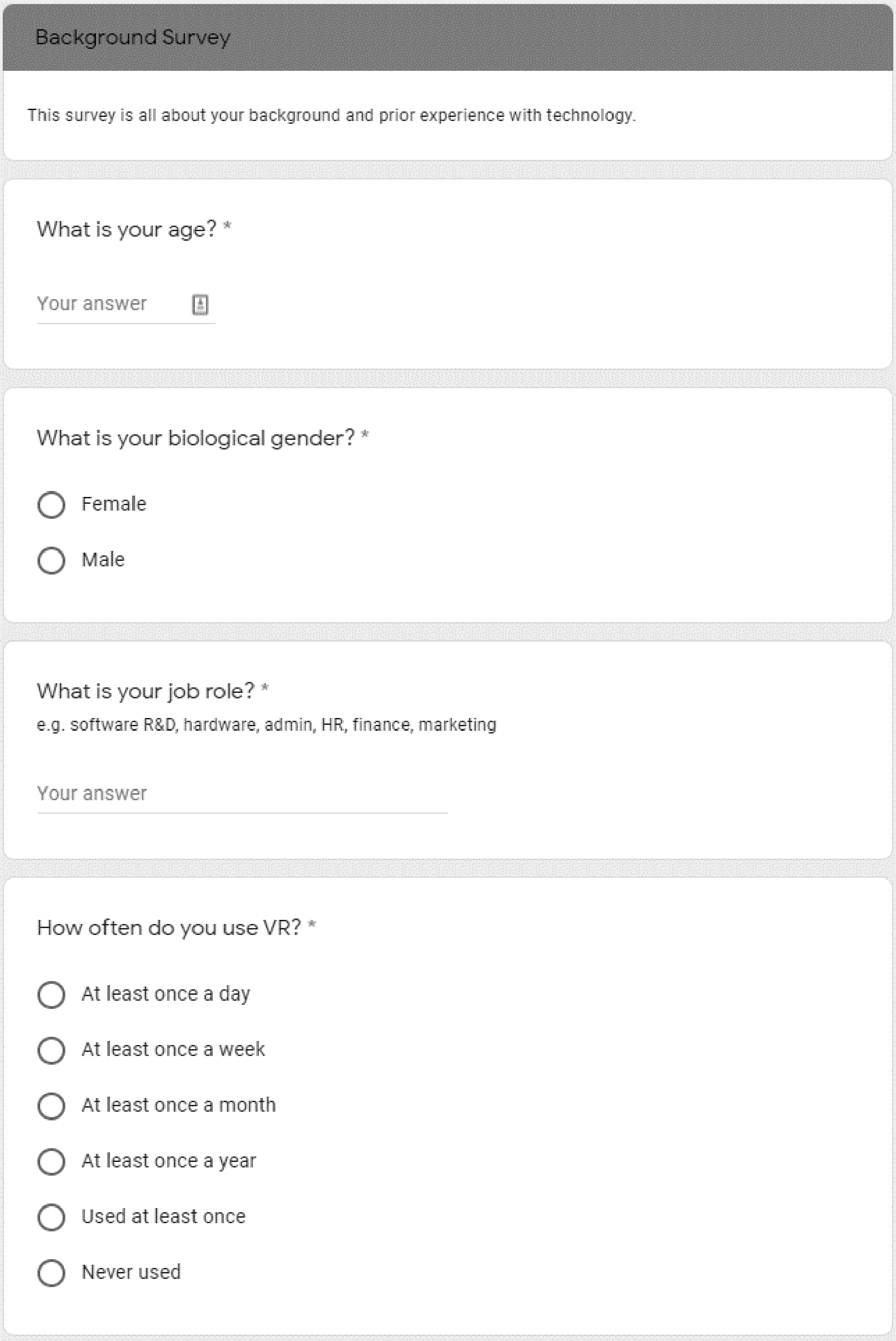
Repeating the studies with greater time frames between repetitions of days or weeks could help prevent learning effects. This may also expose how effective the participants' information recall is, especially with more challenging techniques.

7 Appendices

7.1 Surveys

7.1.1 3D Button Reactions

7.1.1.1 Pre Study



Background Survey

This survey is all about your background and prior experience with technology.

What is your age? *

Your answer

What is your biological gender? *

○ Female

○ Male

What is your job role? *

e.g. software R&D, hardware, admin, HR, finance, marketing

Your answer

How often do you use VR? *

○ At least once a day

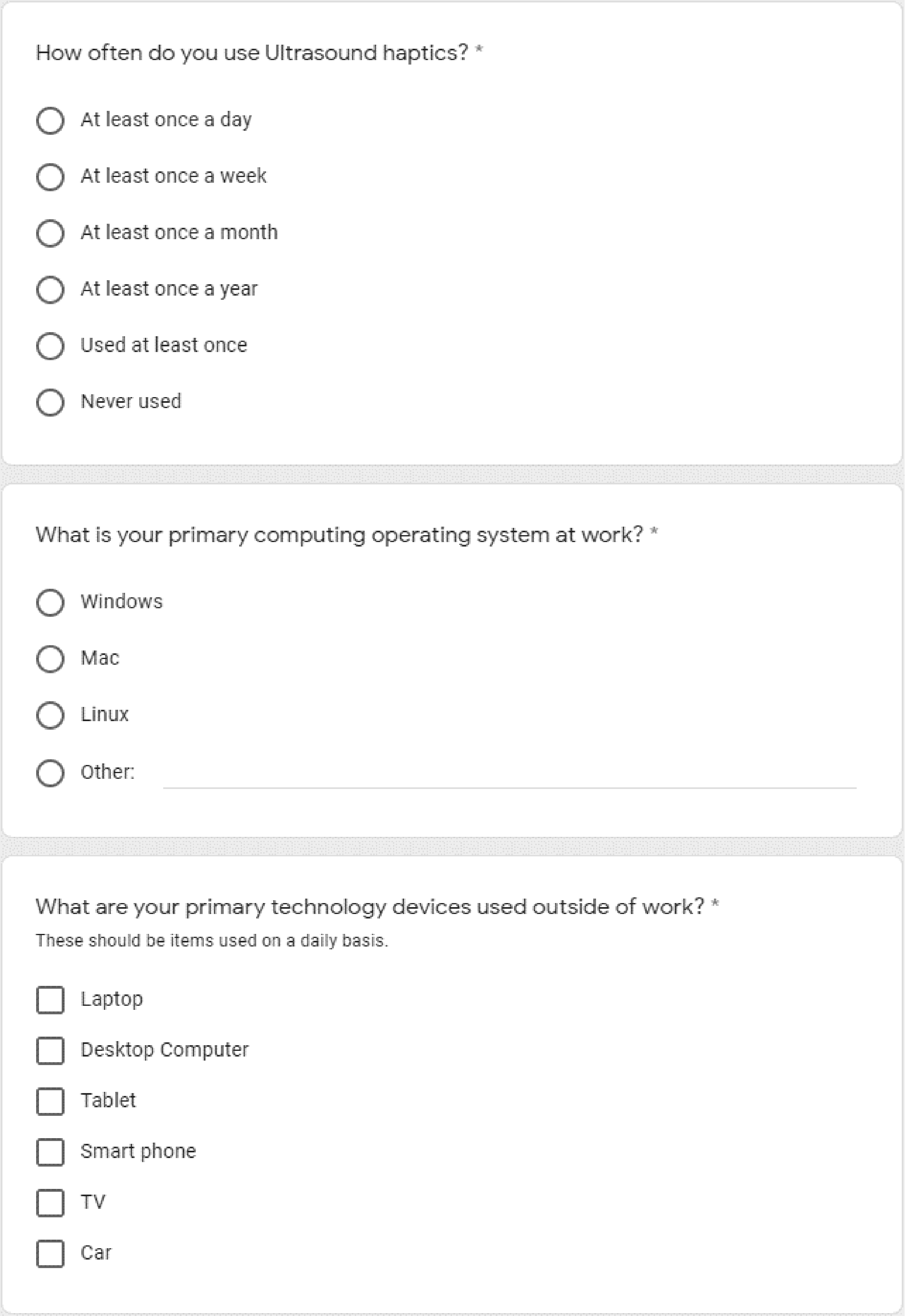
○ At least once a week

○ At least once a month

○ At least once a year

○ Used at least once

○ Never used



How often do you use Ultrasound haptics? *

○ At least once a day
○ At least once a week
○ At least once a month
○ At least once a year
○ Used at least once
○ Never used

What is your primary computing operating system at work? *

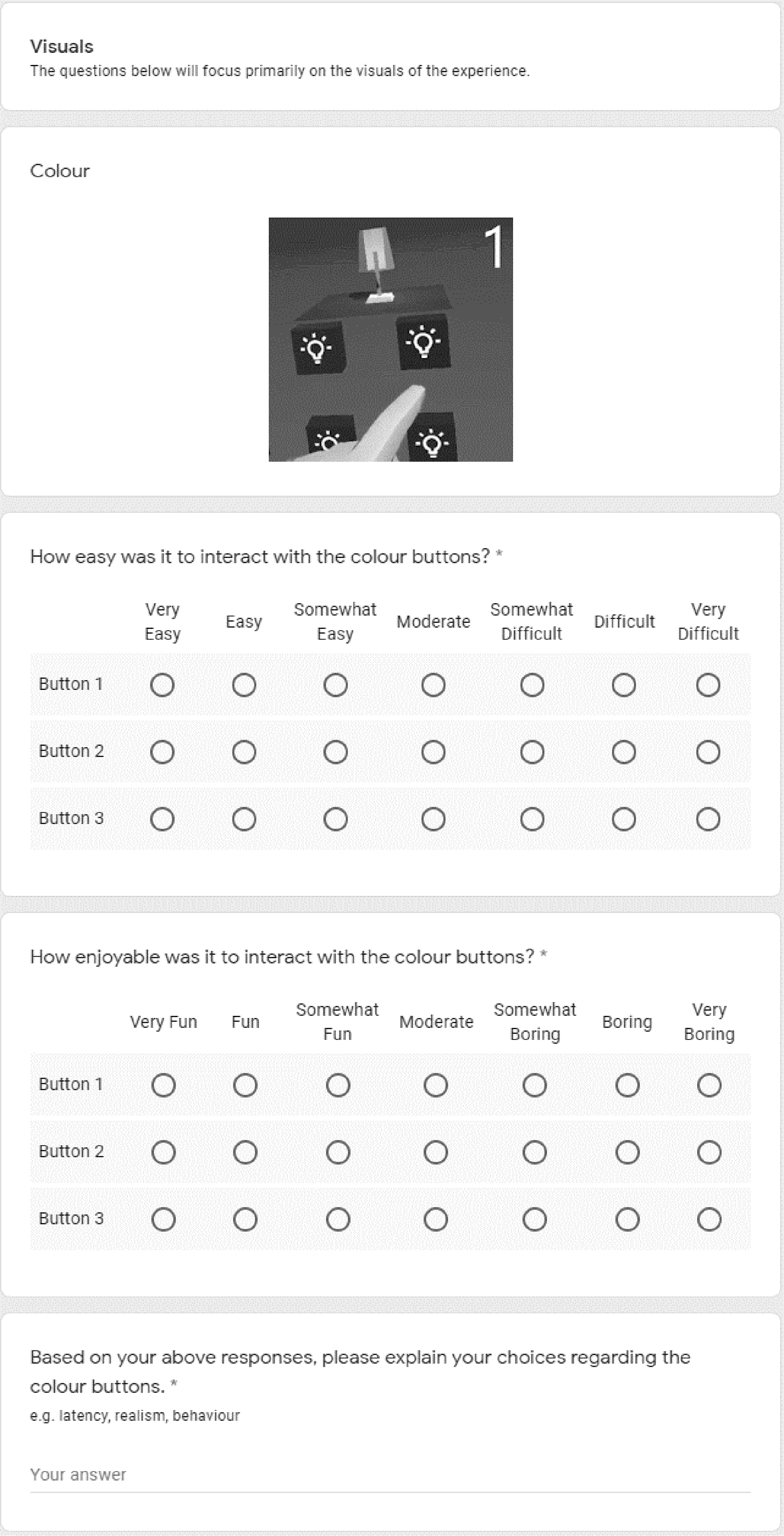
○ Windows
○ Mac
○ Linux
○ Other: ____________________

What are your primary technology devices used outside of work? *

These should be items used on a daily basis.

☐ Laptop
☐ Desktop Computer
☐ Tablet
☐ Smart phone
☐ TV
☐ Car

7.1.1.2 Post Study



Visuals

The questions below will focus primarily on the visuals of the experience.

Colour

How easy was it to interact with the colour buttons? *

	Very Easy	Easy	Somewhat Easy	Moderate	Somewhat Difficult	Difficult	Very Difficult
Button 1	○	○	○	○	○	○	○
Button 2	○	○	○	○	○	○	○
Button 3	○	○	○	○	○	○	○

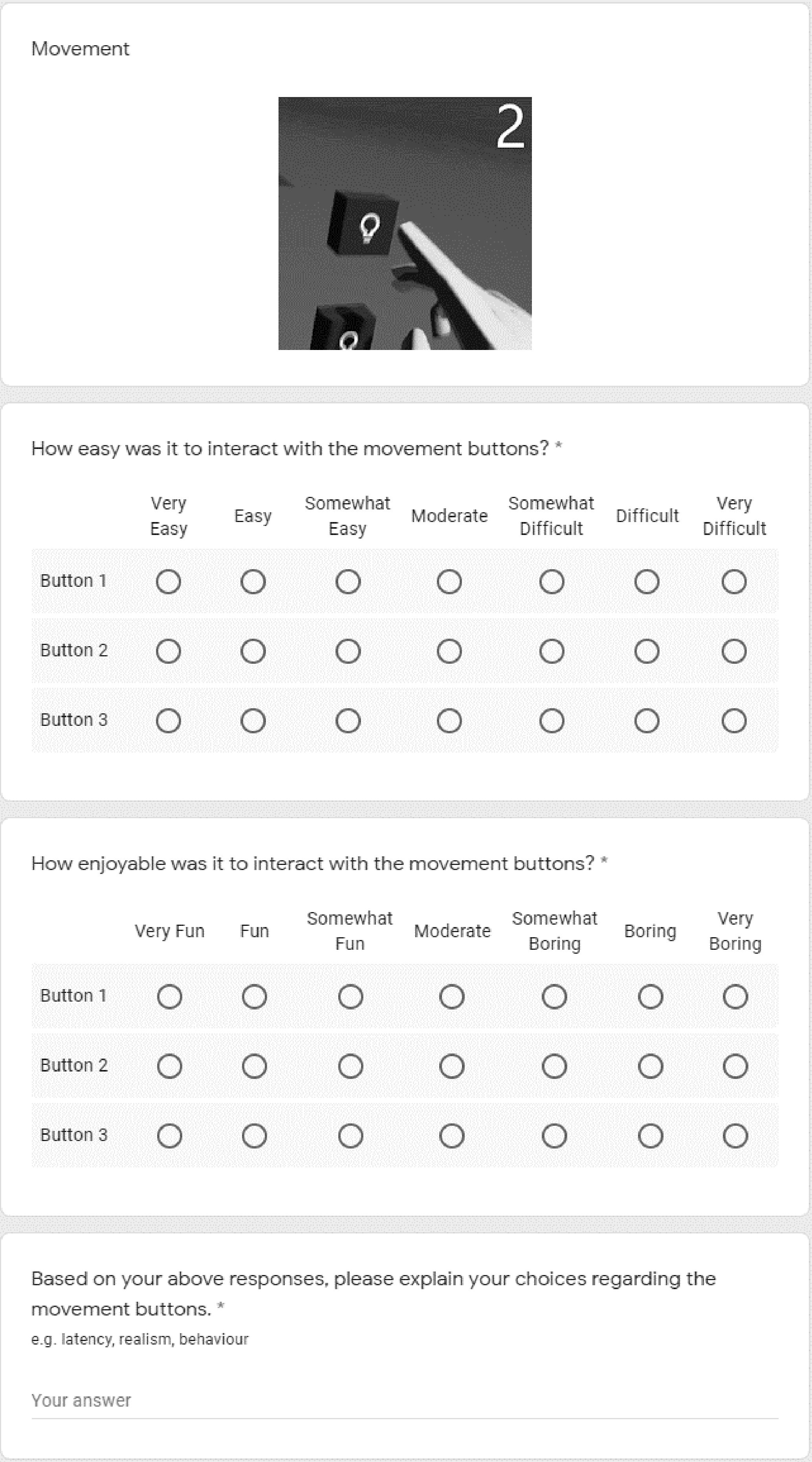
How enjoyable was it to interact with the colour buttons? *

	Very Fun	Fun	Somewhat Fun	Moderate	Somewhat Boring	Boring	Very Boring
Button 1	○	○	○	○	○	○	○
Button 2	○	○	○	○	○	○	○
Button 3	○	○	○	○	○	○	○

Based on your above responses, please explain your choices regarding the colour buttons. *

e.g. latency, realism, behaviour

Your answer



Movement

How easy was it to interact with the movement buttons? *

	Very Easy	Easy	Somewhat Easy	Moderate	Somewhat Difficult	Difficult	Very Difficult
Button 1	○	○	○	○	○	○	○
Button 2	○	○	○	○	○	○	○
Button 3	○	○	○	○	○	○	○

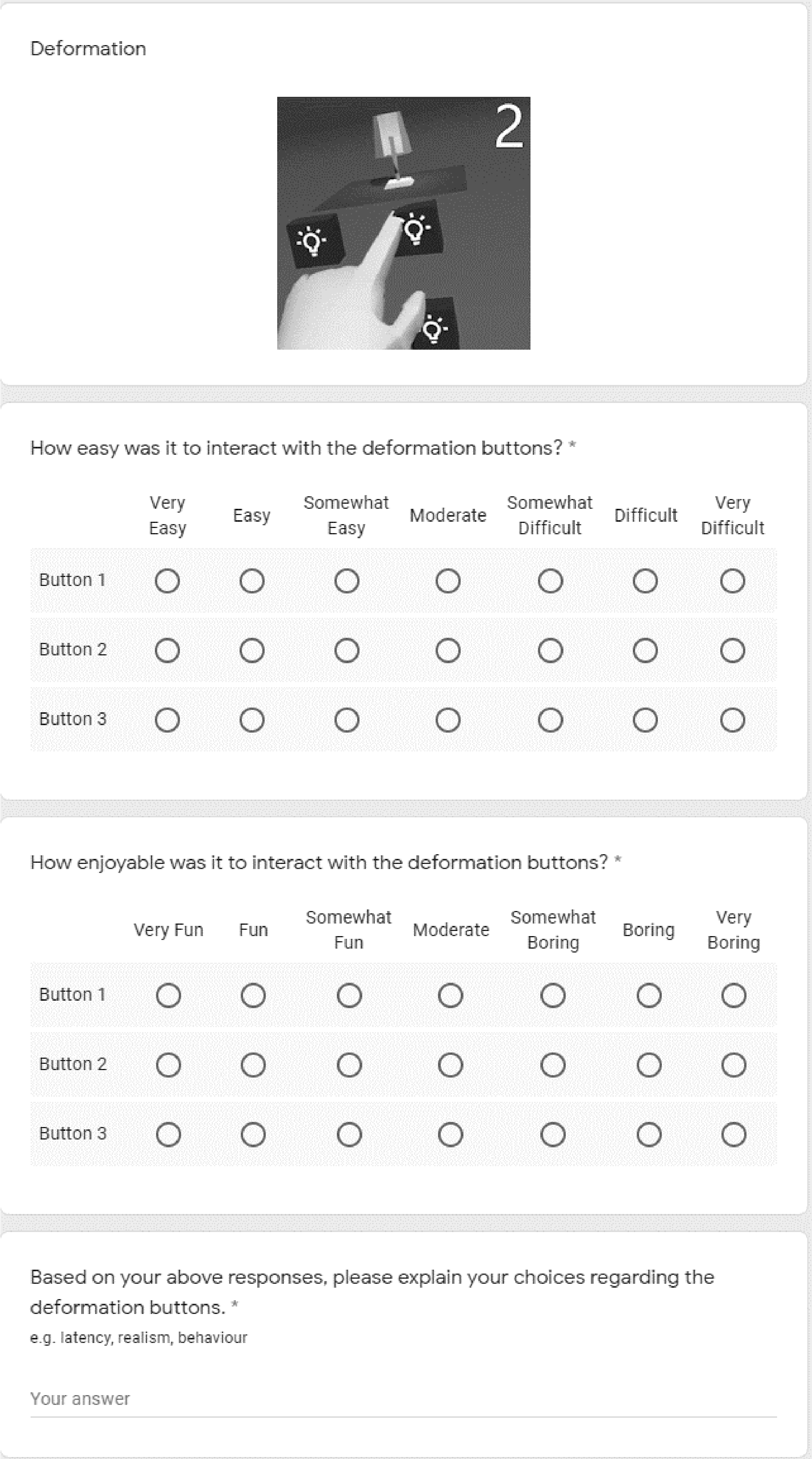
How enjoyable was it to interact with the movement buttons? *

	Very Fun	Fun	Somewhat Fun	Moderate	Somewhat Boring	Boring	Very Boring
Button 1	○	○	○	○	○	○	○
Button 2	○	○	○	○	○	○	○
Button 3	○	○	○	○	○	○	○

Based on your above responses, please explain your choices regarding the movement buttons. *

e.g. latency, realism, behaviour

Your answer



Deformation

How easy was it to interact with the deformation buttons? *

	Very Easy	Easy	Somewhat Easy	Moderate	Somewhat Difficult	Difficult	Very Difficult
Button 1	○	○	○	○	○	○	○
Button 2	○	○	○	○	○	○	○
Button 3	○	○	○	○	○	○	○

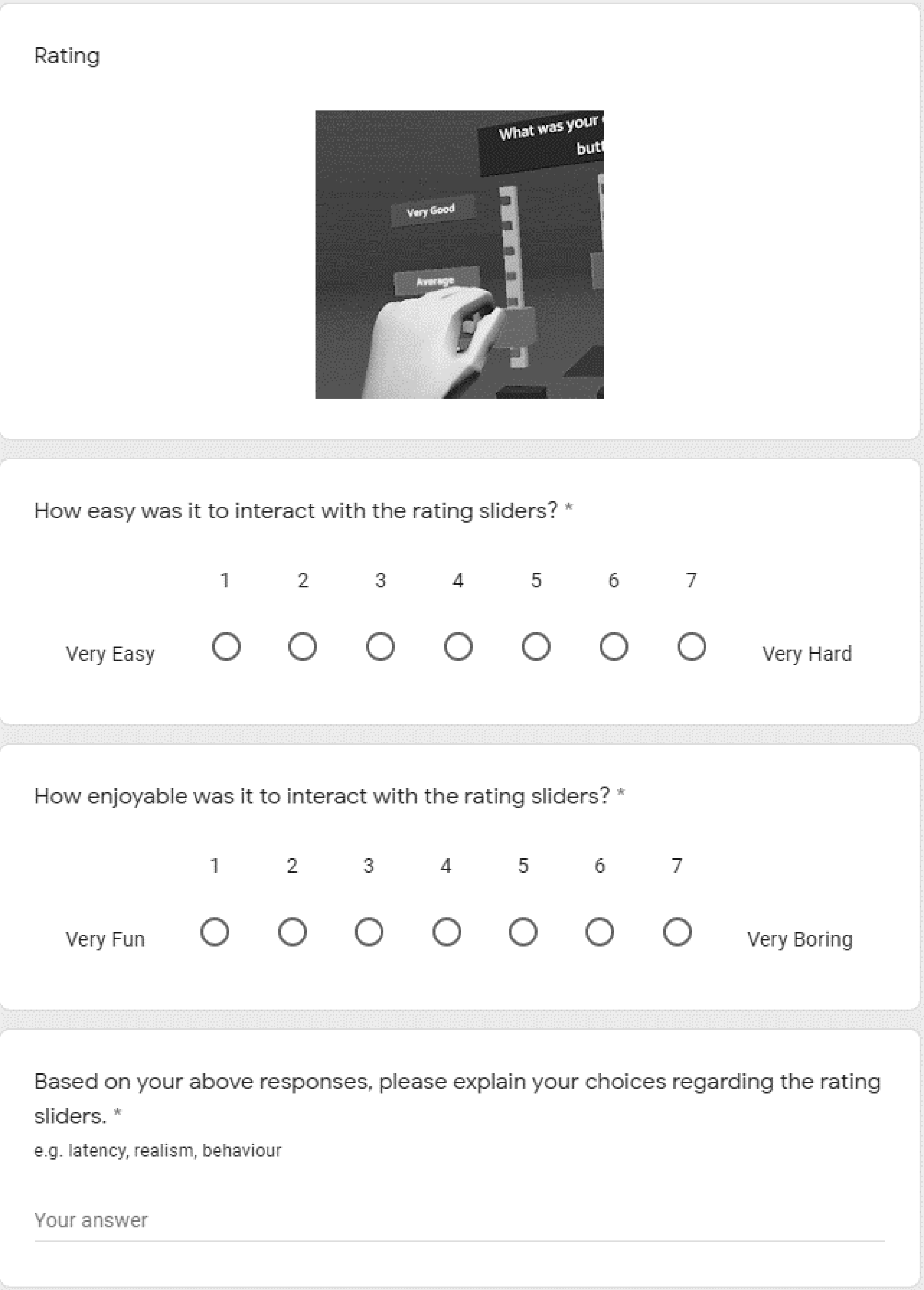
How enjoyable was it to interact with the deformation buttons? *

	Very Fun	Fun	Somewhat Fun	Moderate	Somewhat Boring	Boring	Very Boring
Button 1	○	○	○	○	○	○	○
Button 2	○	○	○	○	○	○	○
Button 3	○	○	○	○	○	○	○

Based on your above responses, please explain your choices regarding the deformation buttons. *

e.g. latency, realism, behaviour

Your answer



Rating

How easy was it to interact with the rating sliders? *

	1	2	3	4	5	6	7	
Very Easy	○	○	○	○	○	○	○	Very Hard

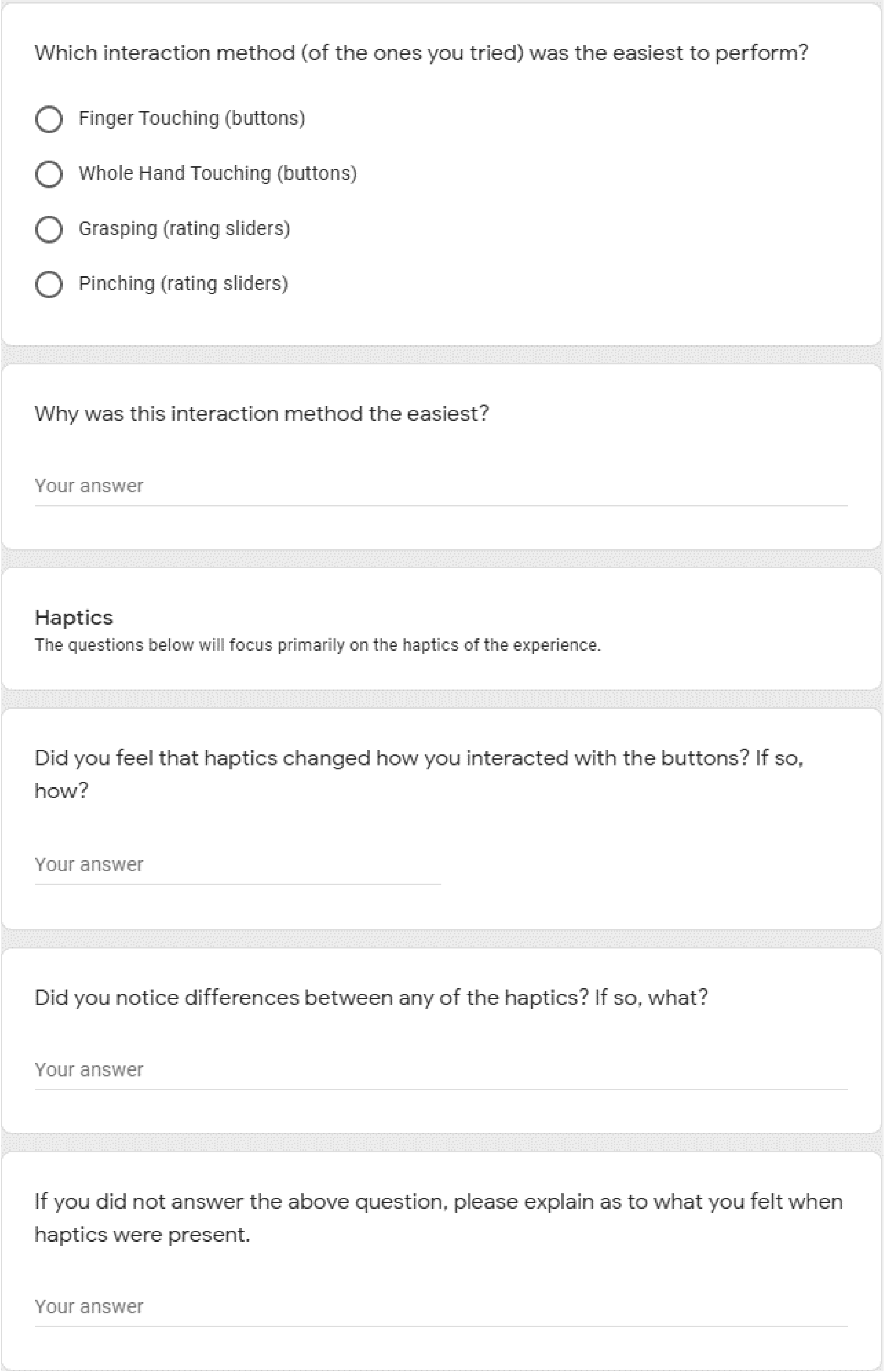
How enjoyable was it to interact with the rating sliders? *

	1	2	3	4	5	6	7	
Very Fun	○	○	○	○	○	○	○	Very Boring

Based on your above responses, please explain your choices regarding the rating sliders. *

e.g. latency, realism, behaviour

Your answer



Which interaction method (of the ones you tried) was the easiest to perform?

- ○ Finger Touching (buttons)
- ○ Whole Hand Touching (buttons)
- ○ Grasping (rating sliders)
- ○ Pinching (rating sliders)

Why was this interaction method the easiest?

Your answer

Haptics

The questions below will focus primarily on the haptics of the experience.

Did you feel that haptics changed how you interacted with the buttons? If so, how?

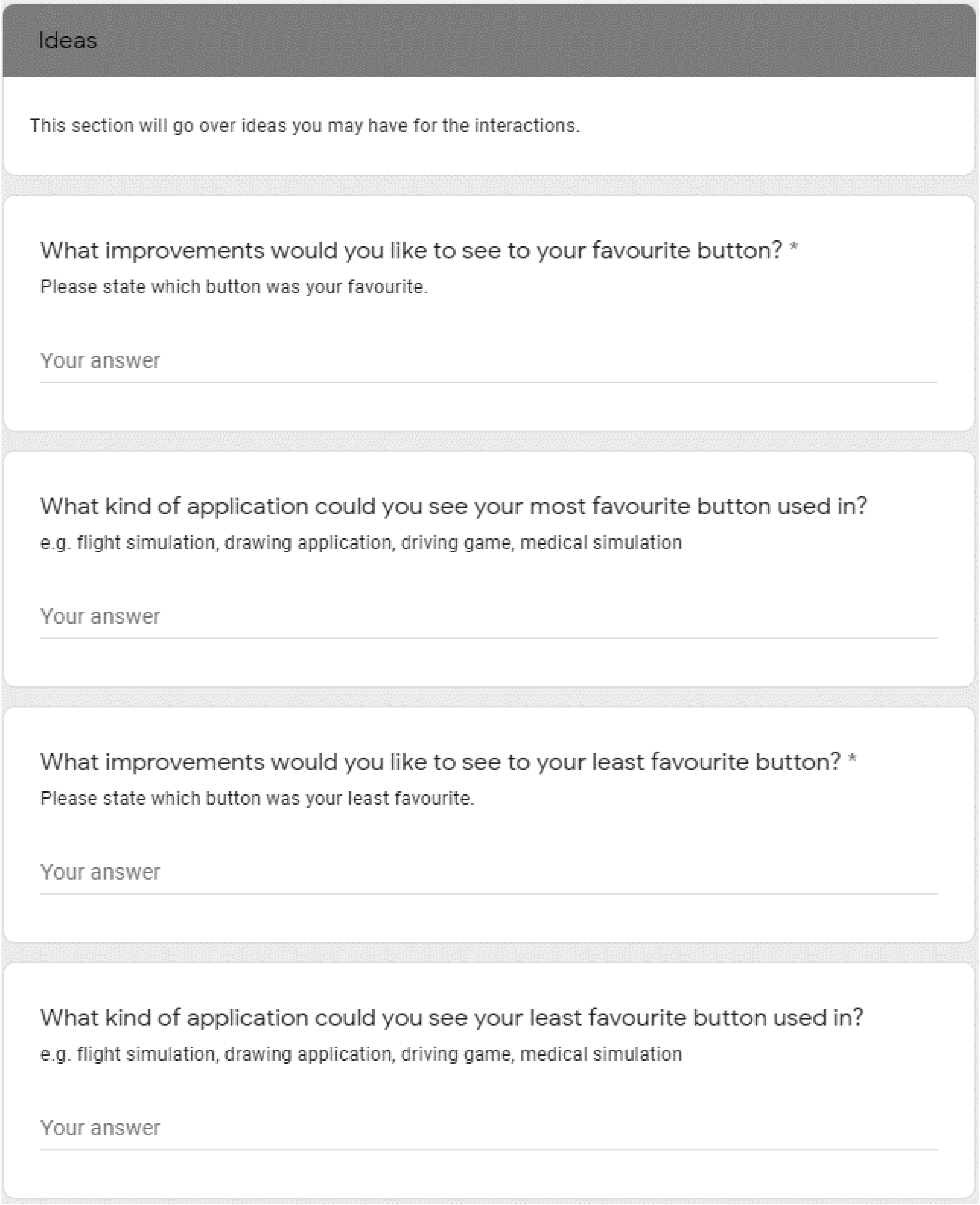
Your answer

Did you notice differences between any of the haptics? If so, what?

Your answer

If you did not answer the above question, please explain as to what you felt when haptics were present.

Your answer



Ideas

This section will go over ideas you may have for the interactions.

What improvements would you like to see to your favourite button? *

Please state which button was your favourite.

Your answer

What kind of application could you see your most favourite button used in?

e.g. flight simulation, drawing application, driving game, medical simulation

Your answer

What improvements would you like to see to your least favourite button? *

Please state which button was your least favourite.

Your answer

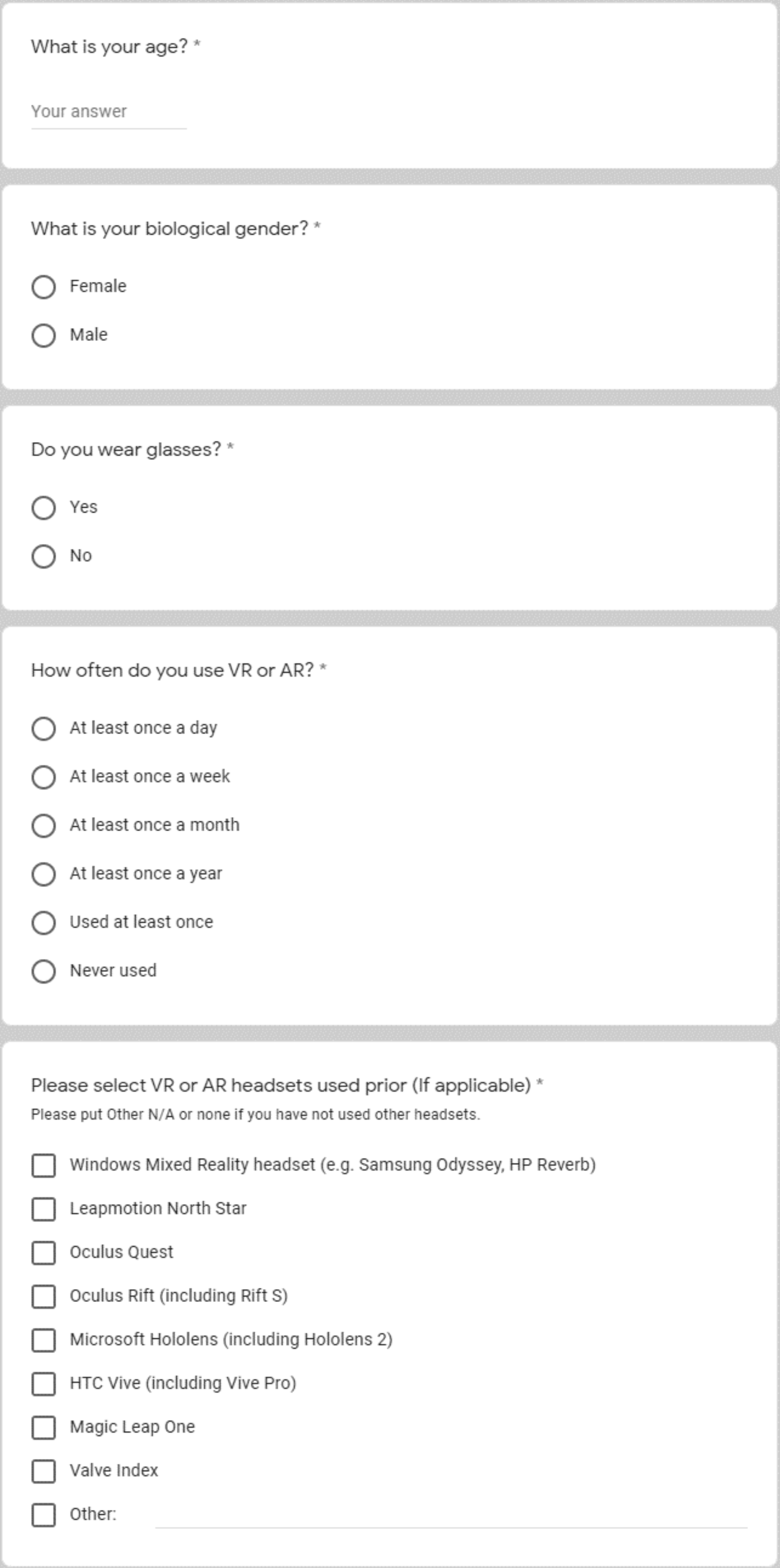
What kind of application could you see your least favourite button used in?

e.g. flight simulation, drawing application, driving game, medical simulation

Your answer

7.1.2 3D Object Interactions

7.1.2.1 Pre Study



What is your age? *

Your answer

What is your biological gender? *

○ Female

○ Male

Do you wear glasses? *

○ Yes

○ No

How often do you use VR or AR? *

○ At least once a day

○ At least once a week

○ At least once a month

○ At least once a year

○ Used at least once

○ Never used

Please select VR or AR headsets used prior (If applicable) *

Please put Other N/A or none if you have not used other headsets.

☐ Windows Mixed Reality headset (e.g. Samsung Odyssey, HP Reverb)

☐ Leapmotion North Star

☐ Oculus Quest

☐ Oculus Rift (including Rift S)

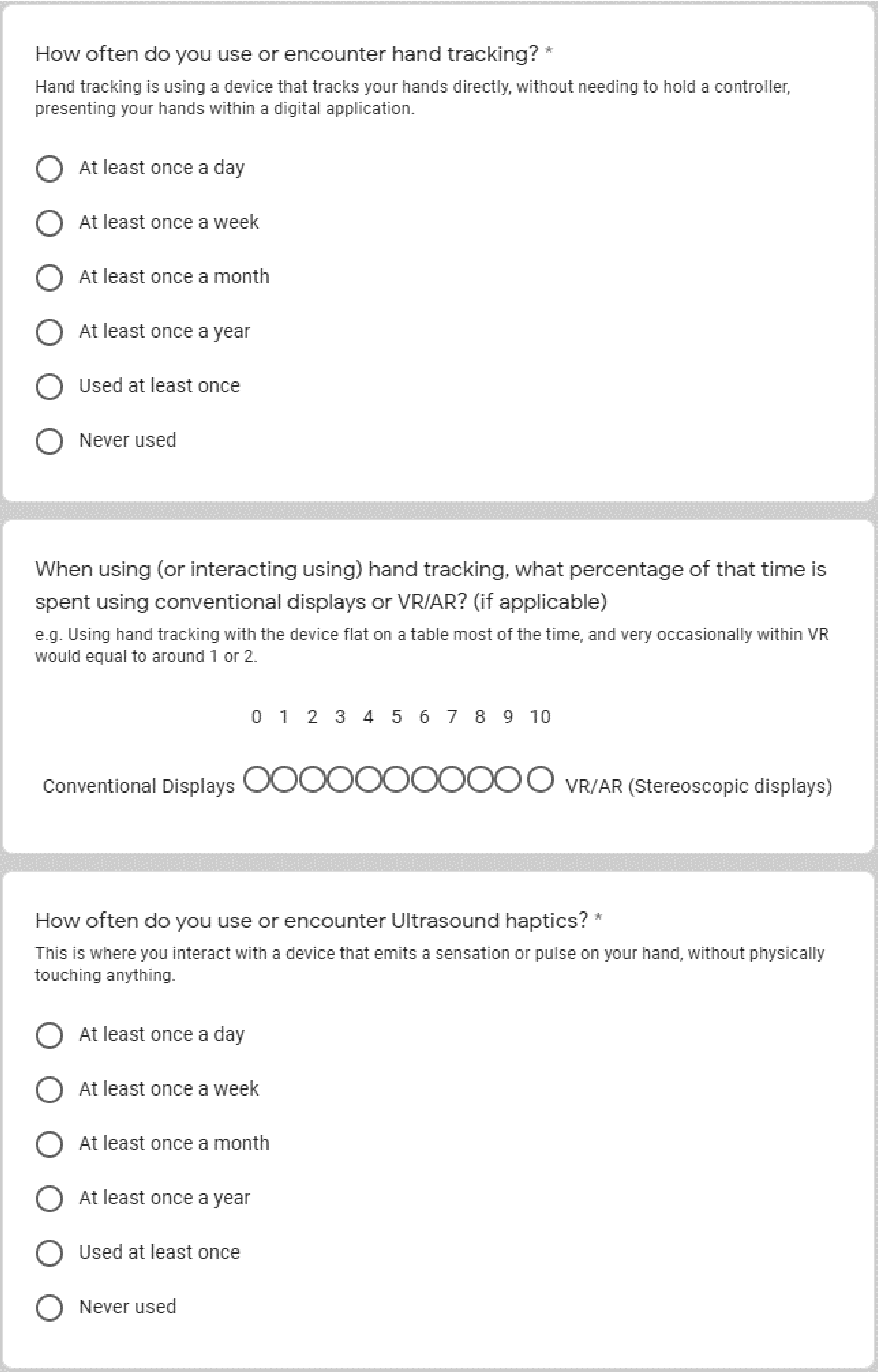
☐ Microsoft Hololens (including Hololens 2)

☐ HTC Vive (including Vive Pro)

☐ Magic Leap One

☐ Valve Index

☐ Other: ________



How often do you use or encounter hand tracking? *

Hand tracking is using a device that tracks your hands directly, without needing to hold a controller, presenting your hands within a digital application.

○ At least once a day

○ At least once a week

○ At least once a month

○ At least once a year

○ Used at least once

○ Never used

When using (or interacting using) hand tracking, what percentage of that time is spent using conventional displays or VR/AR? (if applicable)

e.g. Using hand tracking with the device flat on a table most of the time, and very occasionally within VR would equal to around 1 or 2.

	0	1	2	3	4	5	6	7	8	9	10	
Conventional Displays	○	○	○	○	○	○	○	○	○	○	○	VR/AR (Stereoscopic displays)

How often do you use or encounter Ultrasound haptics? *

This is where you interact with a device that emits a sensation or pulse on your hand, without physically touching anything.

○ At least once a day

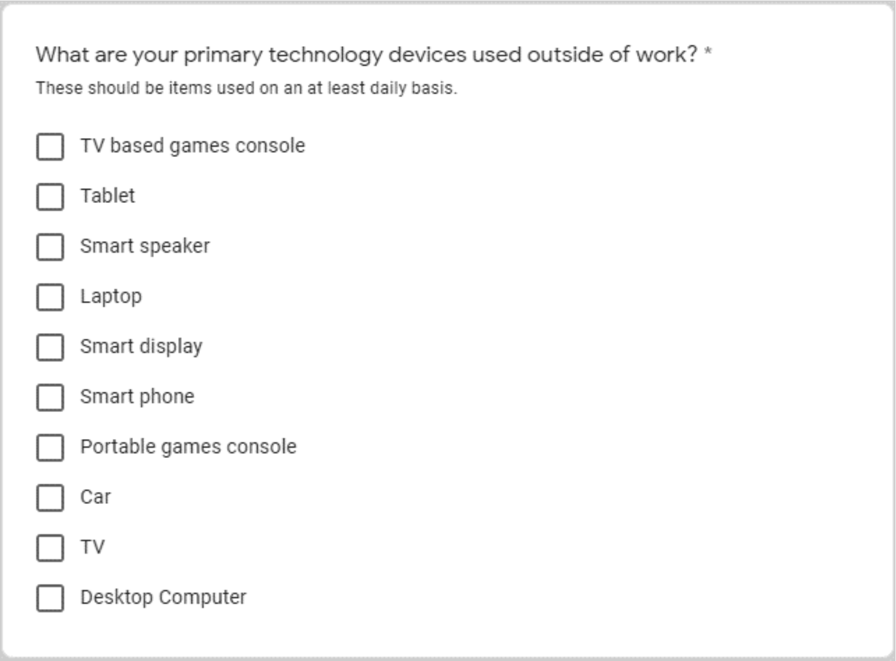
○ At least once a week

○ At least once a month

○ At least once a year

○ Used at least once

○ Never used

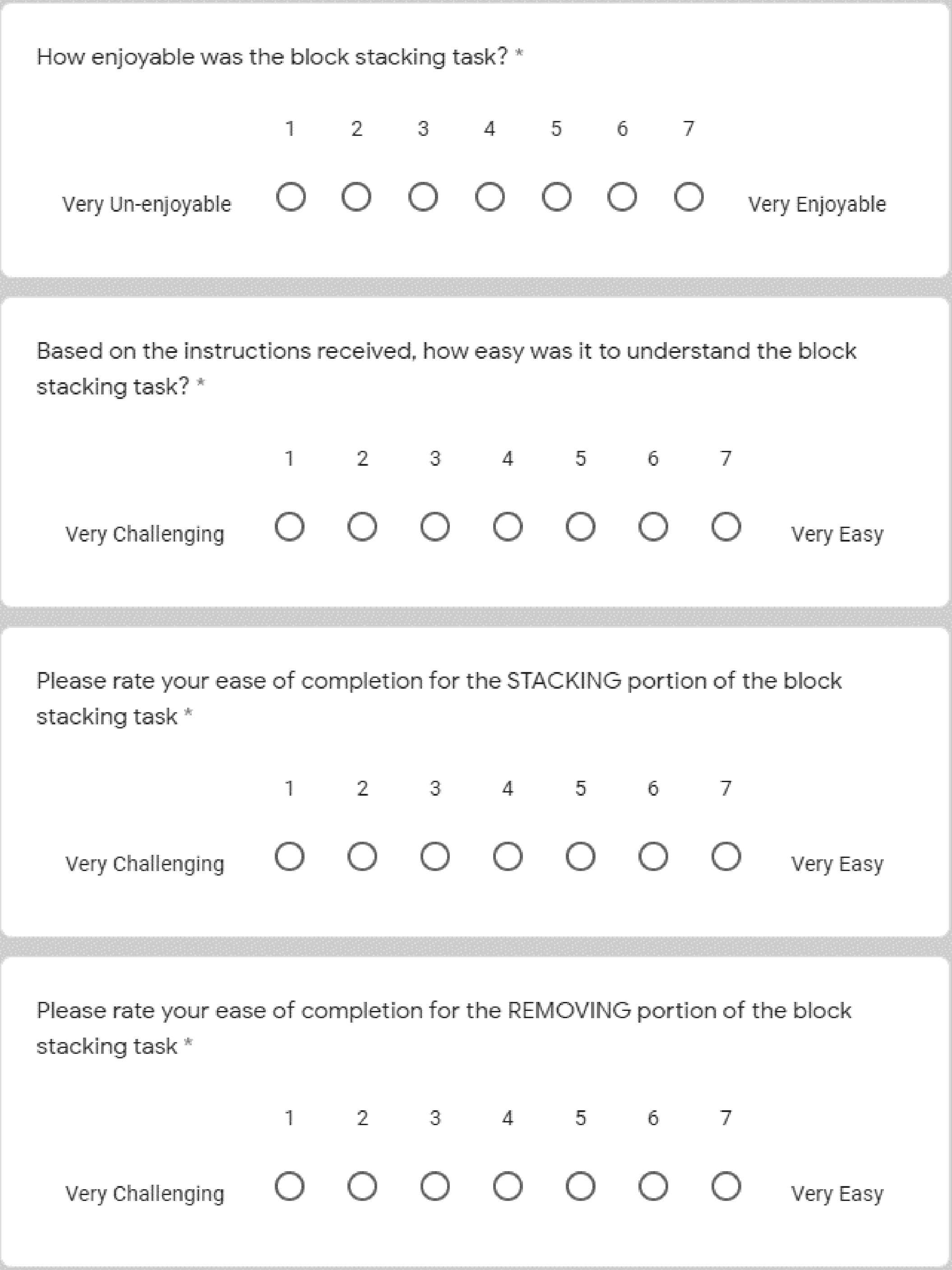


What are your primary technology devices used outside of work? *

These should be items used on an at least daily basis.

☐ TV based games console
☐ Tablet
☐ Smart speaker
☐ Laptop
☐ Smart display
☐ Smart phone
☐ Portable games console
☐ Car
☐ TV
☐ Desktop Computer

7.1.2.2 Post Study



How enjoyable was the block stacking task? *

	1	2	3	4	5	6	7	
Very Un-enjoyable	○	○	○	○	○	○	○	Very Enjoyable

Based on the instructions received, how easy was it to understand the block stacking task? *

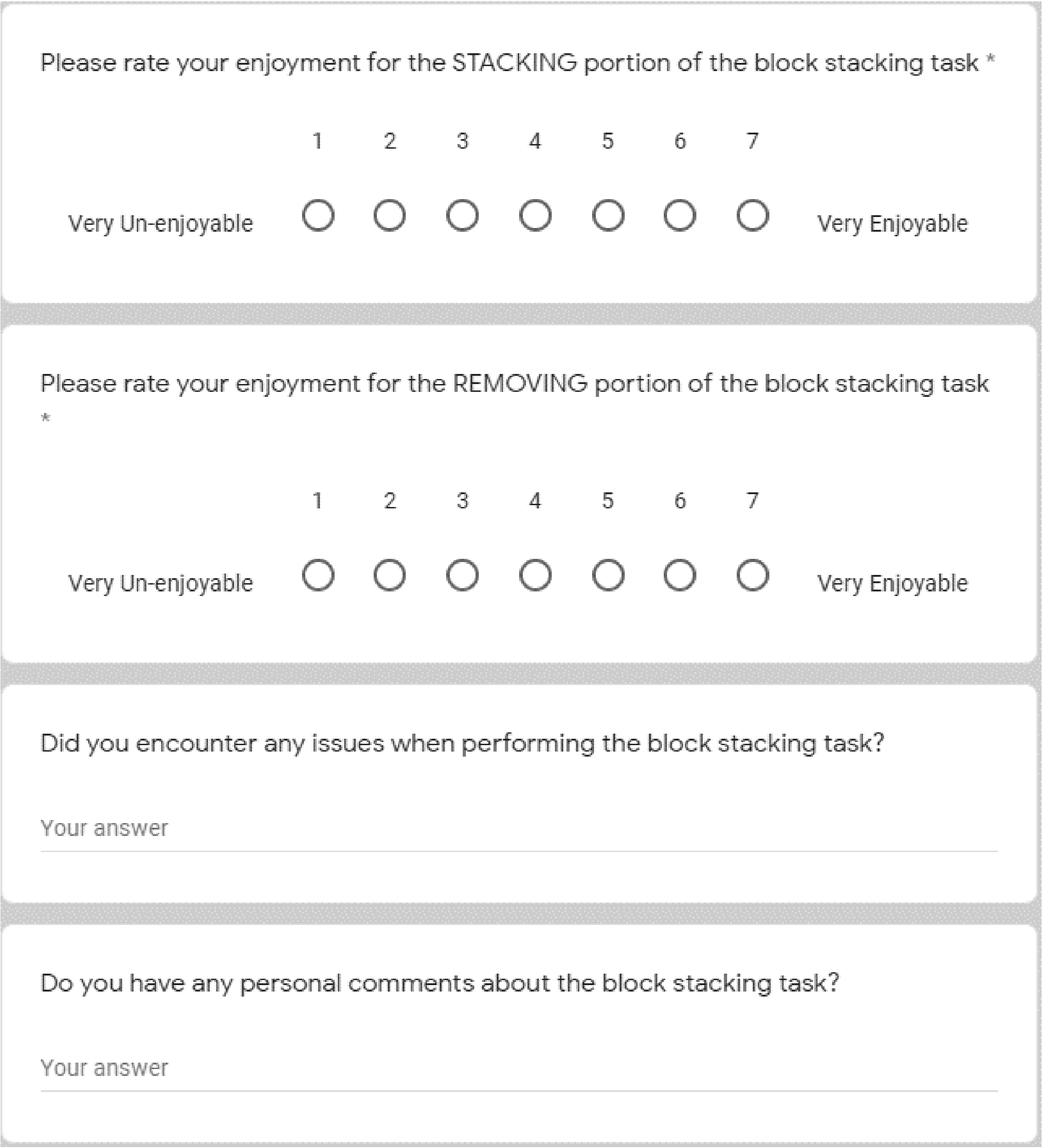
	1	2	3	4	5	6	7	
Very Challenging	○	○	○	○	○	○	○	Very Easy

Please rate your ease of completion for the STACKING portion of the block stacking task *

	1	2	3	4	5	6	7	
Very Challenging	○	○	○	○	○	○	○	Very Easy

Please rate your ease of completion for the REMOVING portion of the block stacking task *

	1	2	3	4	5	6	7	
Very Challenging	○	○	○	○	○	○	○	Very Easy



Please rate your enjoyment for the STACKING portion of the block stacking task *

	1	2	3	4	5	6	7	
Very Un-enjoyable	○	○	○	○	○	○	○	Very Enjoyable

Please rate your enjoyment for the REMOVING portion of the block stacking task *

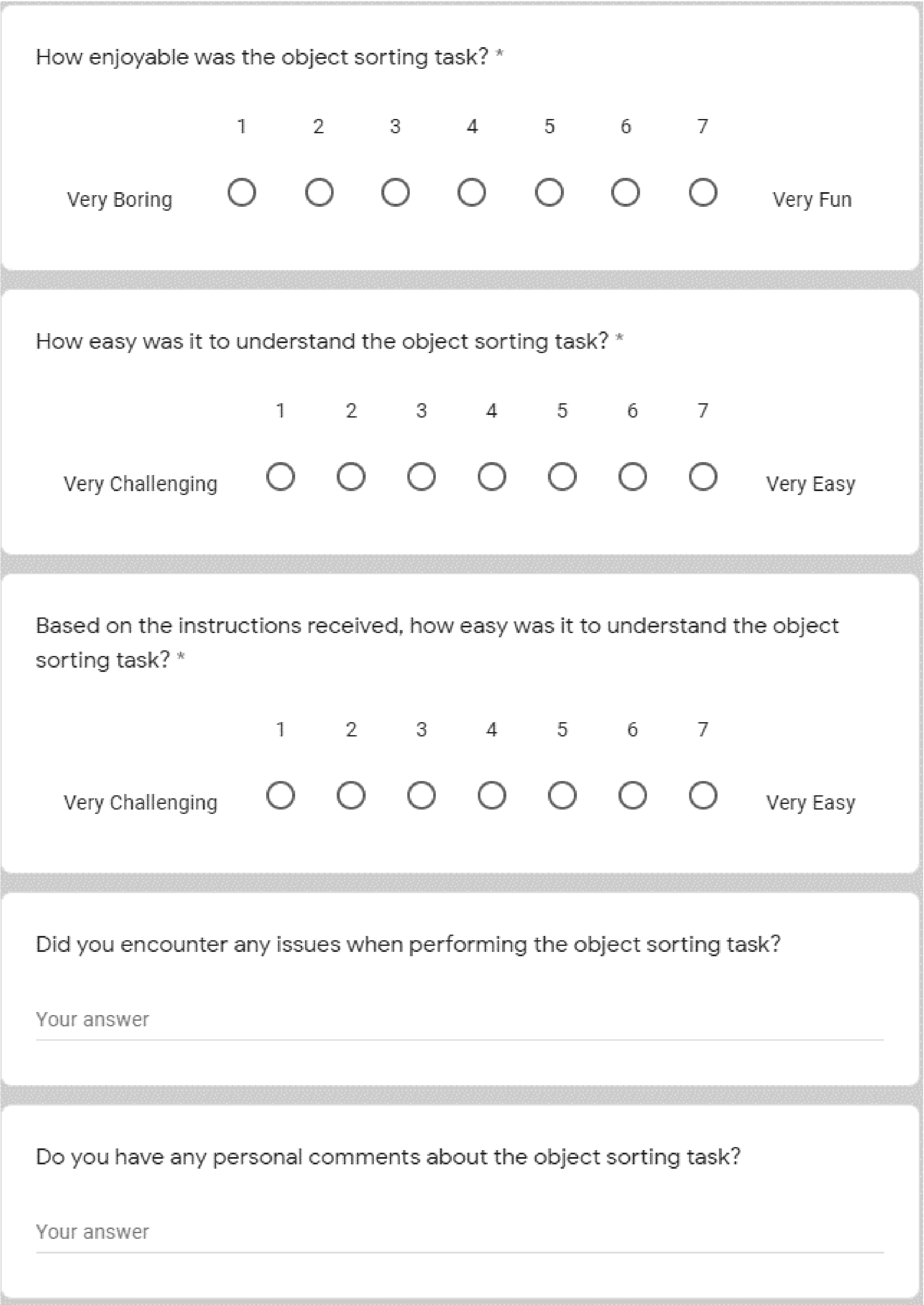
	1	2	3	4	5	6	7	
Very Un-enjoyable	○	○	○	○	○	○	○	Very Enjoyable

Did you encounter any issues when performing the block stacking task?

Your answer

Do you have any personal comments about the block stacking task?

Your answer



How enjoyable was the object sorting task? *

	1	2	3	4	5	6	7	
Very Boring	○	○	○	○	○	○	○	Very Fun

How easy was it to understand the object sorting task? *

	1	2	3	4	5	6	7	
Very Challenging	○	○	○	○	○	○	○	Very Easy

Based on the instructions received, how easy was it to understand the object sorting task? *

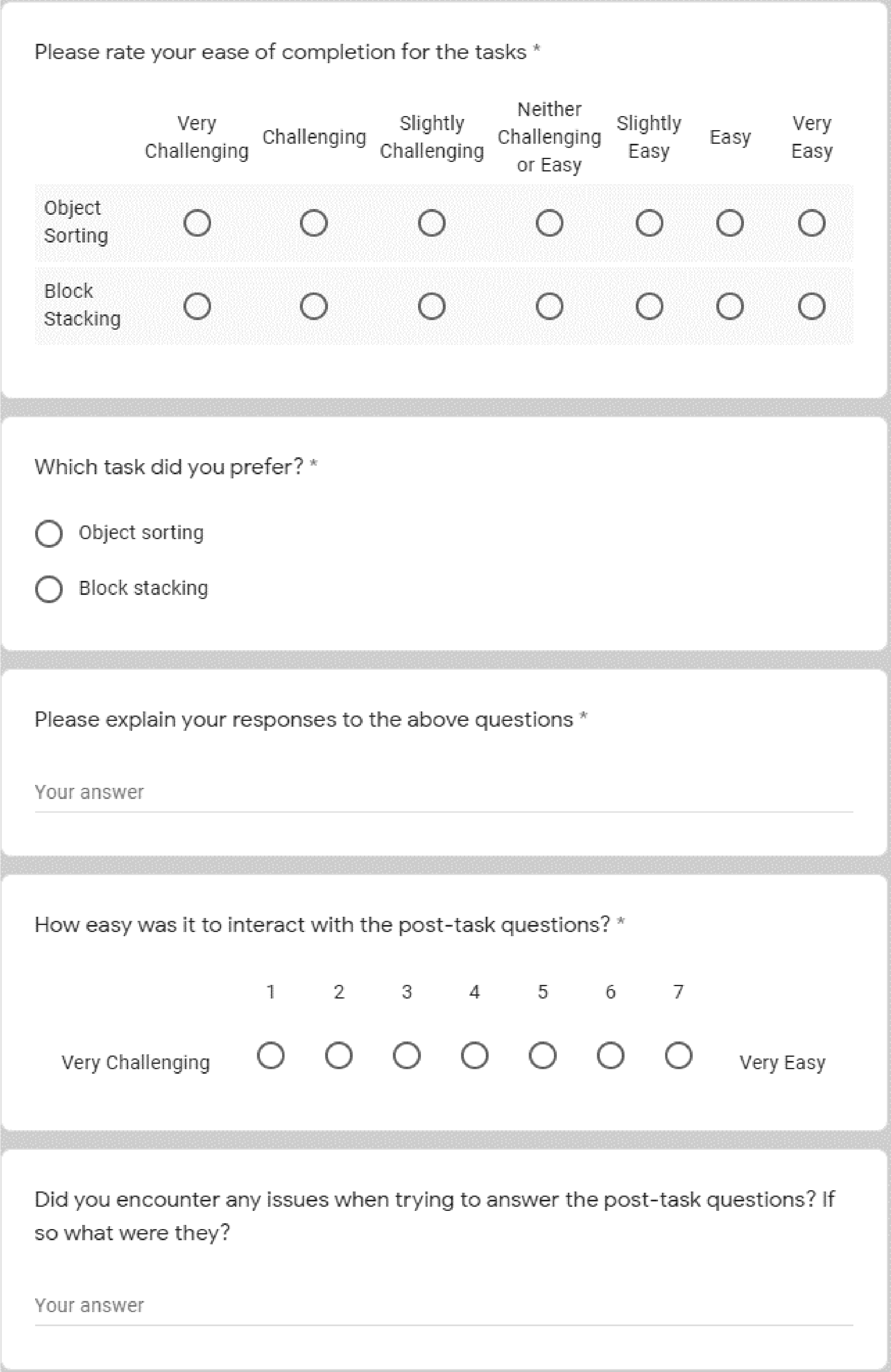
	1	2	3	4	5	6	7	
Very Challenging	○	○	○	○	○	○	○	Very Easy

Did you encounter any issues when performing the object sorting task?

Your answer

Do you have any personal comments about the object sorting task?

Your answer



Please rate your ease of completion for the tasks *

	Very Challenging	Challenging	Slightly Challenging	Neither Challenging or Easy	Slightly Easy	Easy	Very Easy
Object Sorting	○	○	○	○	○	○	○
Block Stacking	○	○	○	○	○	○	○

Which task did you prefer? *

○ Object sorting

○ Block stacking

Please explain your responses to the above questions *

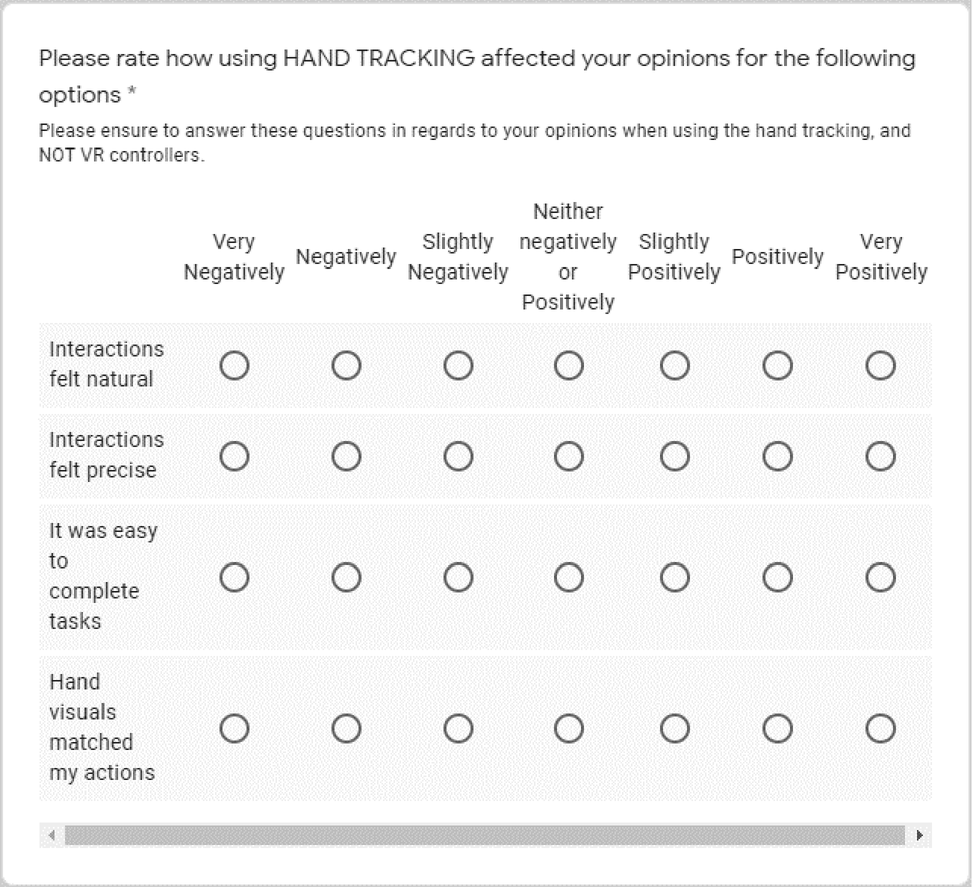
Your answer

How easy was it to interact with the post-task questions? *

	1	2	3	4	5	6	7	
Very Challenging	○	○	○	○	○	○	○	Very Easy

Did you encounter any issues when trying to answer the post-task questions? If so what were they?

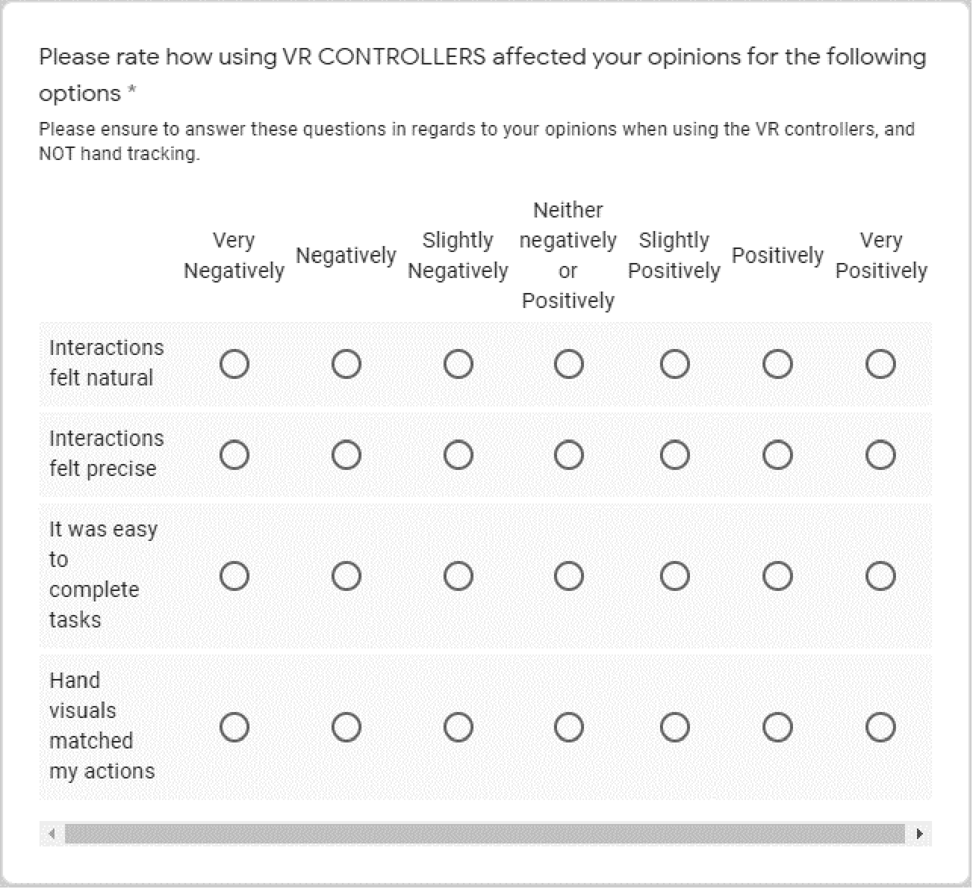
Your answer



Please rate how using HAND TRACKING affected your opinions for the following options *

Please ensure to answer these questions in regards to your opinions when using the hand tracking, and NOT VR controllers.

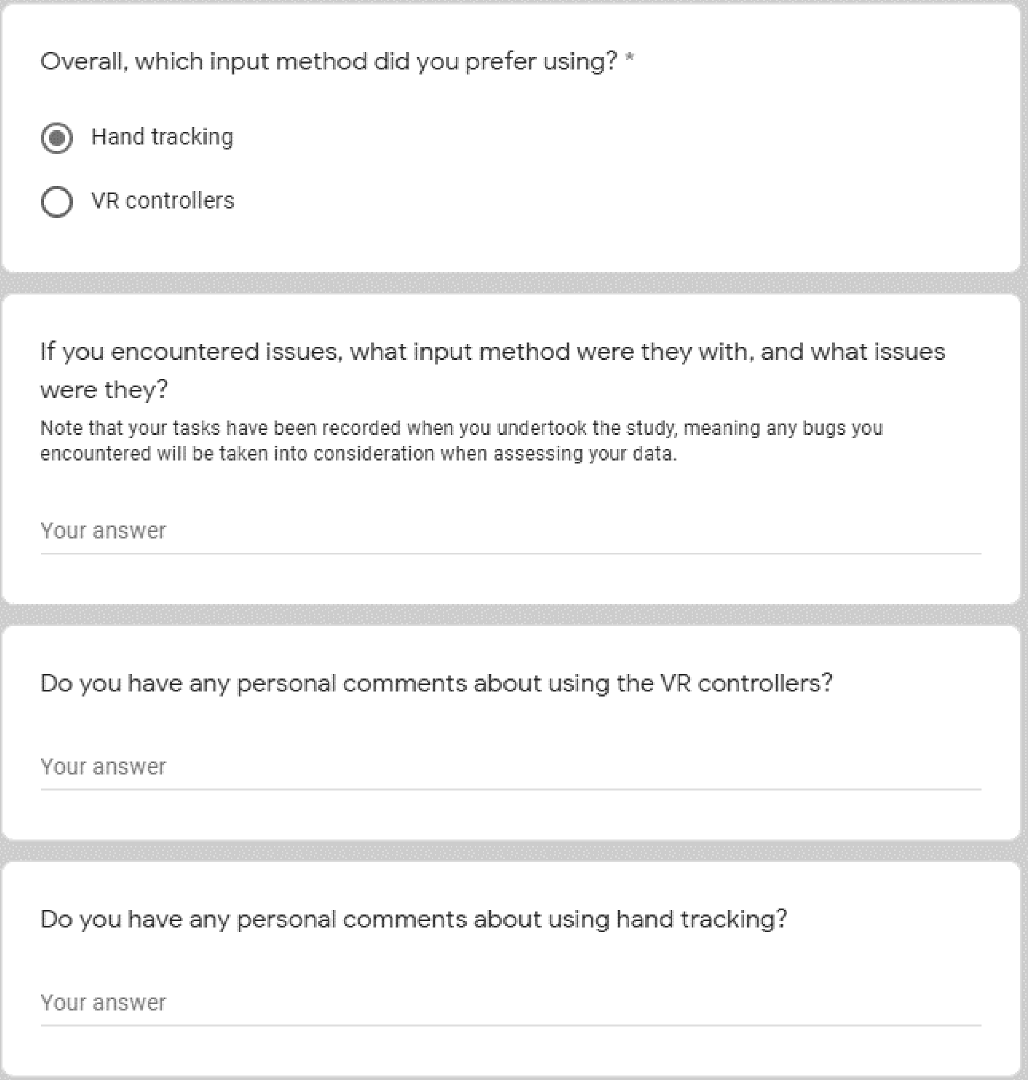
	Very Negatively	Negatively	Slightly Negatively	Neither negatively or Positively	Slightly Positively	Positively	Very Positively
Interactions felt natural	○	○	○	○	○	○	○
Interactions felt precise	○	○	○	○	○	○	○
It was easy to complete tasks	○	○	○	○	○	○	○
Hand visuals matched my actions	○	○	○	○	○	○	○



Please rate how using VR CONTROLLERS affected your opinions for the following options *

Please ensure to answer these questions in regards to your opinions when using the VR controllers, and NOT hand tracking.

	Very Negatively	Negatively	Slightly Negatively	Neither negatively or Positively	Slightly Positively	Positively	Very Positively
Interactions felt natural	○	○	○	○	○	○	○
Interactions felt precise	○	○	○	○	○	○	○
It was easy to complete tasks	○	○	○	○	○	○	○
Hand visuals matched my actions	○	○	○	○	○	○	○



Overall, which input method did you prefer using? *

● Hand tracking

○ VR controllers

If you encountered issues, what input method were they with, and what issues were they?

Note that your tasks have been recorded when you undertook the study, meaning any bugs you encountered will be taken into consideration when assessing your data.

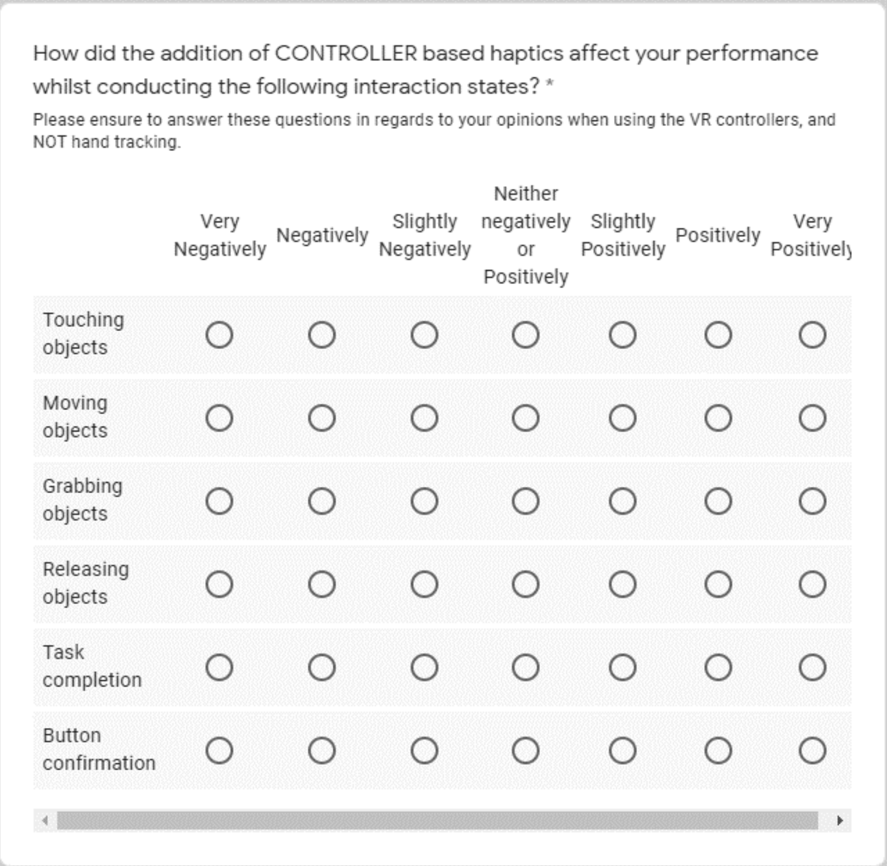
Your answer

Do you have any personal comments about using the VR controllers?

Your answer

Do you have any personal comments about using hand tracking?

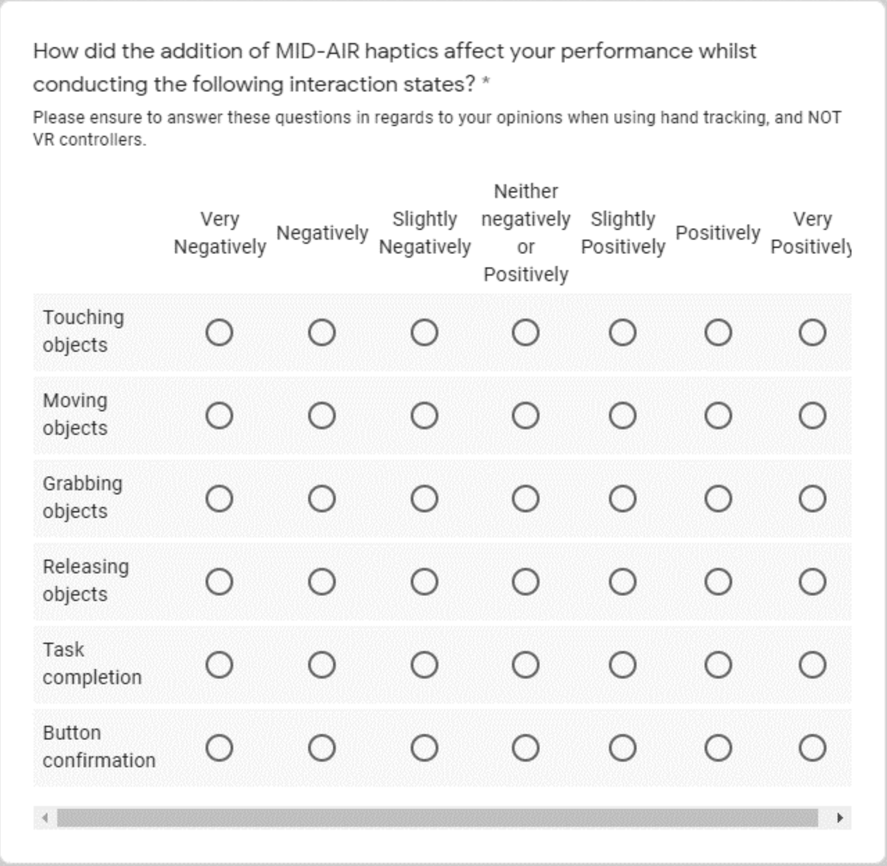
Your answer



How did the addition of CONTROLLER based haptics affect your performance whilst conducting the following interaction states? *

Please ensure to answer these questions in regards to your opinions when using the VR controllers, and NOT hand tracking.

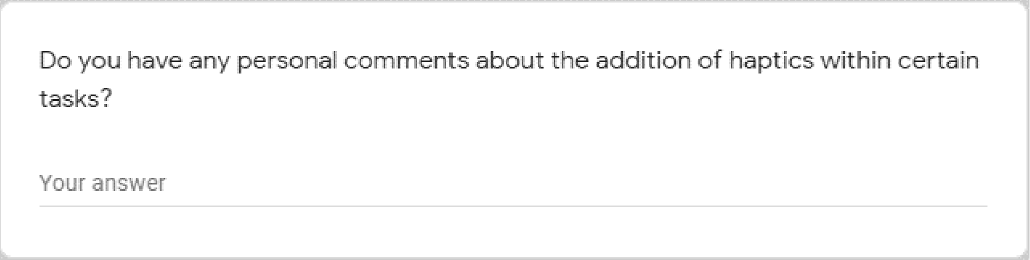
	Very Negatively	Negatively	Slightly Negatively	Neither negatively or Positively	Slightly Positively	Positively	Very Positivel
Touching objects	○	○	○	○	○	○	○
Moving objects	○	○	○	○	○	○	○
Grabbing objects	○	○	○	○	○	○	○
Releasing objects	○	○	○	○	○	○	○
Task completion	○	○	○	○	○	○	○
Button confirmation	○	○	○	○	○	○	○



How did the addition of MID-AIR haptics affect your performance whilst conducting the following interaction states? *

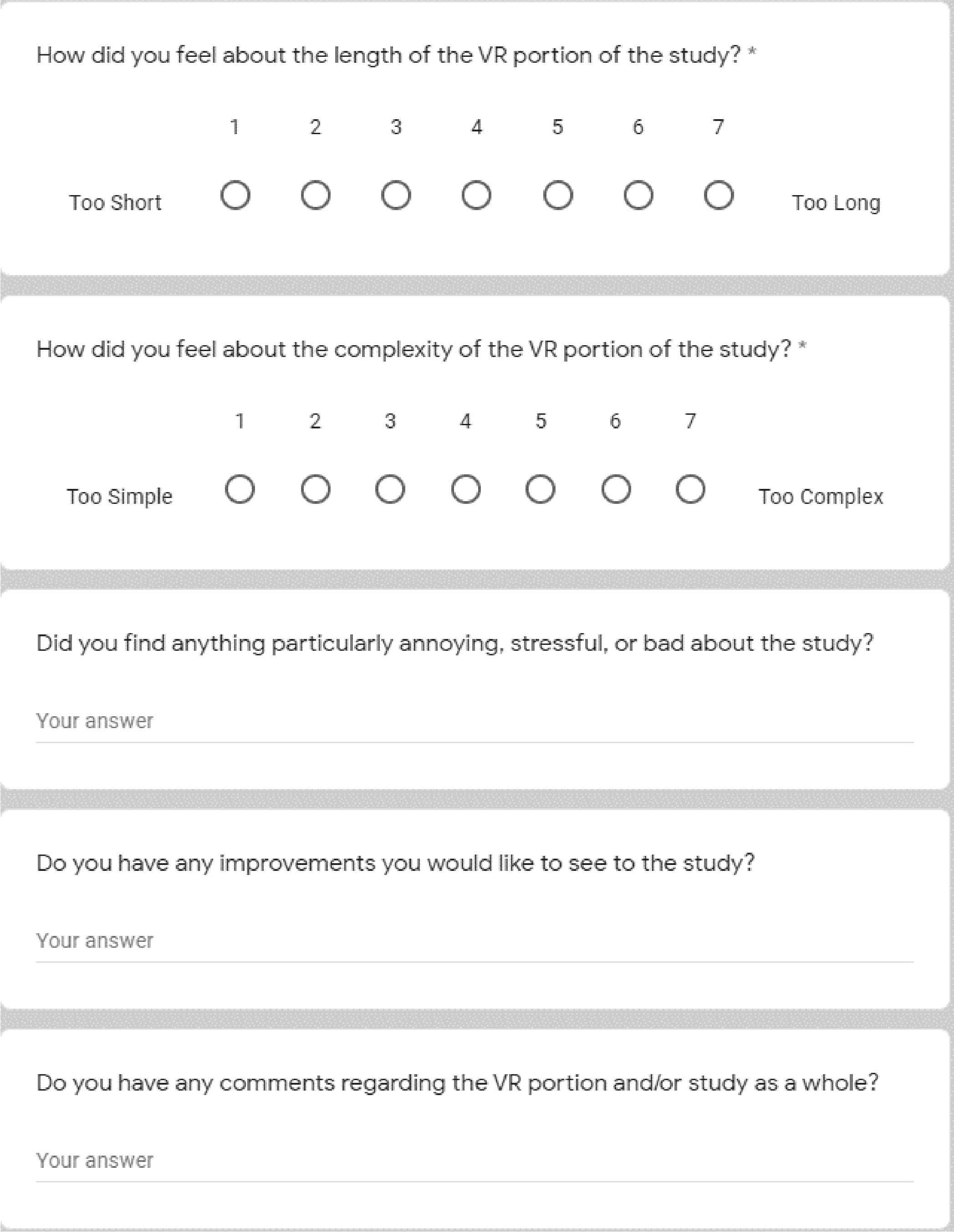
Please ensure to answer these questions in regards to your opinions when using hand tracking, and NOT VR controllers.

	Very Negatively	Negatively	Slightly Negatively	Neither negatively or Positively	Slightly Positively	Positively	Very Positivel
Touching objects	○	○	○	○	○	○	○
Moving objects	○	○	○	○	○	○	○
Grabbing objects	○	○	○	○	○	○	○
Releasing objects	○	○	○	○	○	○	○
Task completion	○	○	○	○	○	○	○
Button confirmation	○	○	○	○	○	○	○



Do you have any personal comments about the addition of haptics within certain tasks?

Your answer



How did you feel about the length of the VR portion of the study? *

	1	2	3	4	5	6	7	
Too Short	○	○	○	○	○	○	○	Too Long

How did you feel about the complexity of the VR portion of the study? *

	1	2	3	4	5	6	7	
Too Simple	○	○	○	○	○	○	○	Too Complex

Did you find anything particularly annoying, stressful, or bad about the study?

Your answer

Do you have any improvements you would like to see to the study?

Your answer

Do you have any comments regarding the VR portion and/or study as a whole?

Your answer

References

Abdouni, A., Djaghloul, M., Thieulin, C., Vargiolu, R., Pailler-Mattei, C. and Zahouani, H., 2017. Biophysical properties of the human finger for touch comprehension: influences of ageing and gender. *Royal Society Open Science*, 4 (8), 170321.

Aditya, K., Chacko, P., Kumari, D., Kumari, D. and Bilgaiyan, S., 2018. Recent Trends in HCI: A survey on Data Glove, LEAP Motion and Microsoft Kinect. *In*: *2018 IEEE International Conference on System, Computation, Automation and Networking (ICSCA)*. Presented at the 2018 IEEE International Conference on System, Computation, Automation and Networking (ICSCA), 1–5.

Ahn, S. J. (Grace), Hahm, J. M. and Johnsen, K., 2019. Feeling the weight of calories: using haptic feedback as virtual exemplars to promote risk perception among young females on unhealthy snack choices. *Media Psychology*, 22 (4), 626–652.

Akay, M., 1998. Force And Touch Feedback For Virtual Reality [Book Reviews]. *Proceedings of the IEEE*, 86 (3), 600–600.

Alzayat, A., Hancock, M. and Nacenta, M. A., 2019. Quantitative Measurement of Tool Embodiment for Virtual Reality Input Alternatives. *In*: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* [online]. New York, NY, USA: Association for Computing Machinery, 1–11. Available from: https://doi.org/10.1145/3290605.3300673 [Accessed 28 Aug 2021].

Apple, 2015. Apple Watch. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=Apple_Watch&oldid=1040197101 [Accessed 23 Aug 2021].

Arafsha, F., Zhang, L., Dong, H. and Saddik, A. E., 2015. Contactless haptic feedback: state of the art. *In*: *2015 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*. Presented at the 2015 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE), 1–6.

Argelaguet, F. and Andujar, C., 2013. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics*, 37 (3), 121–136.

Azmandian, M., Hancock, M., Benko, H., Ofek, E. and Wilson, A. D., 2016. Haptic Retargeting: Dynamic Repurposing of Passive Haptics for Enhanced Virtual Reality Experiences. *In*: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* [online]. New York, NY, USA: Association for Computing Machinery, 1968–1979. Available from: https://doi.org/10.1145/2858036.2858226 [Accessed 27 Aug 2021].

Bachmann, D., Weichert, F. and Rinkenauer, G., 2018. Review of Three-Dimensional Human-Computer Interaction with Focus on the Leap Motion Controller. *Sensors (Basel, Switzerland)* [online], 18 (7). Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6068627/ [Accessed 28 Apr 2020].

Balu, S., 2016. *How does a linear resonant actuator work? - Analog - Technical articles - TI E2E support forums* [online]. Available from: https://e2e.ti.com/blogs_/b/analogwire/posts/how-does-a-linear-resonant-actuator-work [Accessed 23 Aug 2021].

Batmaz, A. U., Machuca, M. D. B., Pham, D. M. and Stuerzlinger, W., 2019. Do Head-Mounted Display Stereo Deficiencies Affect 3D Pointing Tasks in AR and VR? *In*: *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. Presented at the 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 585–592.

Bessa Seixas, M. C., Cardoso, J. C. S. and Dias, M. T. G., 2015. The Leap Motion movement for 2D pointing tasks: Characterisation and comparison to other devices. *In*: *2015 International Conference on Pervasive and Embedded Computing and*

Communication Systems (PECCS). Presented at the 2015 International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS), 15–24.

Boletsis, C. and Cedergren, J. E., 2019. VR Locomotion in the New Era of Virtual Reality: An Empirical Comparison of Prevalent Techniques. *Advances in Human-Computer Interaction*, 2019, e7420781.

Borst, C. W. and Indugula, A. P., 2005. Realistic virtual grasping. *In*: *IEEE Proceedings. VR 2005. Virtual Reality, 2005.* Presented at the IEEE Proceedings. VR 2005. Virtual Reality, 2005., 91–98.

Bowman, D. A., Chen, J., Wingrave, C. A., Lucas, J., Ray, A., Polys, N. F., Li, Q., Haciahmetoglu, Y., Kim, J.-S., Kim, S., Boehringer, R. and Ni, T., 2006. New Directions in 3D User Interfaces. *International Journal of Virtual Reality*, 5 (2), 3–14.

Bowman, D. A., McMahan, R. P. and Ragan, E. D., 2012. Questioning naturalism in 3D user interfaces. *Communications of the ACM*, 55 (9), 78.

Brookes, J., 2021. *UXF - Unity Experiment Framework* [online]. Immersive Cognition. Available from: https://github.com/immersivecognition/unity-experiment-framework [Accessed 22 Aug 2021].

Cabrera Hidalgo, J. C., Arévalo Delgado, J. D., Bykbaev, V. R., Robles Bykbaev, Y. and Pesántez Coyago, T., 2018. Serious game to improve fine motor skills using Leap Motion. *In*: *2018 Congreso Argentino de Ciencias de la Informática y Desarrollos de Investigación (CACIDI)*. Presented at the 2018 Congreso Argentino de Ciencias de la Informática y Desarrollos de Investigación (CACIDI), 1–5.

Caggianese, G., Gallo, L. and Neroni, P., 2019. The Vive Controllers vs. Leap Motion for Interactions in Virtual Environments: A Comparative Evaluation. *In*: De Pietro, G., Gallo, L., Howlett, R. J., Jain, L. C., and Vlacic, L., eds. *Intelligent Interactive Multimedia Systems and Services*. Cham: Springer International Publishing, 24–33.

Carter, T., Seah, S. A., Long, B., Drinkwater, B. and Subramanian, S., 2013a. UltraHaptics: multi-point mid-air haptic feedback for touch surfaces. *In*: *Proceedings of the 26th annual ACM symposium on User interface software and technology* [online]. St. Andrews, Scotland, United Kingdom: Association for Computing Machinery, 505–514. Available from: https://doi.org/10.1145/2501988.2502018 [Accessed 28 Apr 2020].

Carter, T., Seah, S. A., Long, B., Drinkwater, B. and Subramanian, S., 2013b. UltraHaptics: multi-point mid-air haptic feedback for touch surfaces. *In*: *Proceedings of the 26th annual ACM symposium on User interface software and technology* [online]. St. Andrews, Scotland, United Kingdom: Association for Computing Machinery, 505–514. Available from: https://doi.org/10.1145/2501988.2502018 [Accessed 28 Apr 2020].

Chan, A., Halevi, T. and Memon, N., 2015. Leap Motion Controller for Authentication via Hand Geometry and Gestures. *In*: Tryfonas, T. and Askoxylakis, I., eds. *Human Aspects of Information Security, Privacy, and Trust*. Cham: Springer International Publishing, 13–22.

Choi, I., Ofek, E., Benko, H., Sinclair, M. and Holz, C., 2018. CLAW: A Multifunctional Handheld Haptic Controller for Grasping, Touching, and Triggering in Virtual Reality. *In*: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* [online]. New York, NY, USA: Association for Computing Machinery, 1–13. Available from: https://doi.org/10.1145/3173574.3174228 [Accessed 27 May 2021].

Christ, R. D. and Wernli, R. L., 2014. Chapter 19 - Manipulators. *In*: Christ, R. D. and Wernli, R. L., eds. *The ROV Manual (Second Edition)* [online]. Oxford: Butterworth-

Heinemann, 503–534. Available from: https://www.sciencedirect.com/science/article/pii/B9780080982885000191 [Accessed 24 Aug 2021].

Clark, R., 2021. *ultraleap/PlayRecorder* [online]. Ultraleap. Available from: https://github.com/ultraleap/PlayRecorder [Accessed 1 Jun 2021].

Claypool, M., Claypool, K. and Damaa, F., 2006. The effects of frame rate and resolution on users playing first person shooter games. *In*: *Multimedia Computing and Networking 2006* [online]. Presented at the Multimedia Computing and Networking 2006, International Society for Optics and Photonics, 607101. Available from: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/6071/607101/The-effects-of-frame-rate-and-resolution-on-users-playing/10.1117/12.648609.short [Accessed 28 Apr 2020].

Coburn, J. Q., Freeman, I. and Salmon, J. L., 2017. A Review of the Capabilities of Current Low-Cost Virtual Reality Technology and Its Potential to Enhance the Design Process. *Journal of Computing and Information Science in Engineering* [online], 17 (031013). Available from: https://doi.org/10.1115/1.4036921 [Accessed 25 May 2021].

Cognitive3D, 2021. *Analytics for VR, AR, MR* [online]. Available from: https://cognitive3d.com/ [Accessed 18 Jun 2021].

Conti, F., Barbagli, F., Balaniuk, R., Halg, M., Lu, C., Morris, D., Sentis, L., Warren, J., Khatib, O. and Salisbury, K., 2003. The CHAI libraries. *In*: *Proceedings of Eurohaptics 2003*. Dublin, Ireland, 496–500.

Corvinus, J., 2021. *jcorvinus/HandshakeVR* [online]. Available from: https://github.com/jcorvinus/HandshakeVR [Accessed 27 May 2021].

Cournia, N., Smith, J. D. and Duchowski, A. T., 2003. Gaze- vs. hand-based pointing in virtual environments. *In*: *CHI '03 Extended Abstracts on Human Factors in Computing Systems* [online]. New York, NY, USA: Association for Computing Machinery, 772–773. Available from: https://doi.org/10.1145/765891.765982 [Accessed 5 Sep 2021].

De Paolis, L. T. and De Luca, V., 2020. The impact of the input interface in a virtual environment: the Vive controller and the Myo armband. *Virtual Reality*, 24 (3), 483–502.

Dexta Robotics, 2021. *Dexta Robotics - Touch the Untouchable* [online]. Available from: https://www.dextarobotics.com/ [Accessed 25 May 2021].

Ens, B., Byagowi, A., Han, T., Hincapié-Ramos, J. D. and Irani, P., 2016. Combining Ring Input with Hand Tracking for Precise, Natural Interaction with Spatial Analytic Interfaces. *In*: *Proceedings of the 2016 Symposium on Spatial User Interaction* [online]. New York, NY, USA: Association for Computing Machinery, 99–102. Available from: https://doi.org/10.1145/2983310.2985757 [Accessed 13 Sep 2021].

Erden, F. and Çetin, A. E., 2014. Hand gesture based remote control system using infrared sensors and a camera. *IEEE Transactions on Consumer Electronics*, 60 (4), 675–680.

Falcao, C., Lemos, A. C. and Soares, M., 2015. Evaluation of Natural User Interface: A Usability Study Based on the Leap Motion Device. *Procedia Manufacturing*, 3, 5490–5495.

Francone, A., Huang, J. M., Ma, J., Tsao, T.-C., Rosen, J. and Hubschman, J.-P., 2019. The Effect of Haptic Feedback on Efficiency and Safety During Preretinal Membrane Peeling Simulation. *Translational Vision Science & Technology*, 8 (4), 2–2.

Frier, W., Ablart, D., Chilles, J., Long, B., Giordano, M., Obrist, M. and Subramanian, S., 2018. Using Spatiotemporal Modulation to Draw Tactile Patterns in Mid-Air. *In*: Prattichizzo, D., Shinoda, H., Tan, H. Z., Ruffaldi, E., and Frisoli, A., eds. *Haptics:*

Science, Technology, and Applications. Cham: Springer International Publishing, 270–281.

Fröhner, J., Salvietti, G., Beckerle, P. and Prattichizzo, D., 2019. Can Wearable Haptic Devices Foster the Embodiment of Virtual Limbs? *IEEE Transactions on Haptics*, 12 (3), 339–349.

Gaffary, Y. and Lécuyer, A., 2018. The Use of Haptic and Tactile Information in the Car to Improve Driving Safety: A Review of Current Technologies. *Frontiers in ICT*, 5, 5.

Gallo, L., 2013. A study on the degrees of freedom in touchless interaction. *In*: *SIGGRAPH Asia 2013 Technical Briefs* [online]. New York, NY, USA: Association for Computing Machinery, 1–4. Available from: https://doi.org/10.1145/2542355.2542390 [Accessed 27 Aug 2021].

Ganguly, A., Rashidi, G. and Mombaur, K., 2021. Comparison of the Performance of the Leap Motion ControllerTM with a Standard Marker-Based Motion Capture System. *Sensors*, 21 (5), 1750.

Georgiou, O., Jeffrey, C., Chen, Z., Xiao Tong, B., Hei Chan, S., Yang, B., Harwood, A. and Carter, T., 2018. Touchless Haptic Feedback for VR Rhythm Games. *In*: *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. Presented at the 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 553–554.

Guna, J., Jakus, G., Pogačnik, M., Tomažič, S. and Sodnik, J., 2014. An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking. *Sensors*, 14 (2), 3702–3720.

Gunawardane, P. D. S. H. and Medagedara, N. T., 2017. Comparison of hand gesture inputs of leap motion controller amp; data glove in to a soft finger. *In*: *2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*. Presented at the 2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), 62–68.

Guzsvinecz, T., Szucs, V. and Sik-Lanyi, C., 2019. Suitability of the Kinect Sensor and Leap Motion Controller—A Literature Review. *Sensors (Basel, Switzerland)* [online], 19 (5). Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6427122/ [Accessed 26 May 2021].

Hachisu, T., Cirio, G., Marchal, M., Lécuyer, A. and Kajimoto, H., 2011. Pseudo-haptic feedback augmented with visual and tactile vibrations. *In*: *2011 IEEE International Symposium on VR Innovation*. Presented at the 2011 IEEE International Symposium on VR Innovation, 327–328.

Hadi, R. and Valenzuela, A., 2020. Good Vibrations: Consumer Responses to Technology-Mediated Haptic Feedback. *Journal of Consumer Research*, 47 (2), 256–271.

Hagelsteen, K., Johansson, R., Ekelund, M., Bergenfelz, A. and Anderberg, M., 2019. Performance and perception of haptic feedback in a laparoscopic 3D virtual reality simulator. *Minimally Invasive Therapy & Allied Technologies*, 28 (5), 309–316.

Hald, K., 2013. Low-cost 3DUI using hand tracking and controllers. *In*: *2013 IEEE Symposium on 3D User Interfaces (3DUI)*. Presented at the 2013 IEEE Symposium on 3D User Interfaces (3DUI), 205–206.

Han, I. and Black, J. B., 2011. Incorporating haptic feedback in simulation for learning physics. *Computers & Education*, 57 (4), 2281–2290.

Holderied, H., 2017. *Evaluation of Interaction Concepts in Virtual Reality Applications* [online]. Gesellschaft für Informatik, Bonn. Available from: http://dl.gi.de/handle/20.500.12116/4030 [Accessed 28 Apr 2020].

Hoppe, M., Knierim, P., Kosch, T., Funk, M., Futami, L., Schneegass, S., Henze, N., Schmidt, A. and Machulla, T., 2018. VRHapticDrones: Providing Haptics in Virtual Reality through Quadcopters. *In*: *Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia* [online]. New York, NY, USA: Association for

Computing Machinery, 7–18. Available from: https://doi.org/10.1145/3282894.3282898 [Accessed 21 Aug 2021].

Horowitz, K., 2014. *Skeletal Tracking 101: Getting Started with the Bone API* [online]. Leap Motion Blog. Available from: http://blog.leapmotion.com/skeletal-tracking-101-getting-started-with-the-bone-api-and-rigged-hands/ [Accessed 28 Apr 2020].

Hoshi, T., Takahashi, M., Iwamoto, T. and Shinoda, H., 2010. Noncontact Tactile Display Based on Radiation Pressure of Airborne Ultrasound. *IEEE Transactions on Haptics*, 3 (3), 155–165.

HTC, 2016a. HTC Vive. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=HTC_Vive&oldid=1022940984 [Accessed 25 May 2021].

HTC, 2016b. *VIVE Controller | VIVE* [online]. Available from: https://www.vive.com/uk/accessory/controller/www.vive.com/uk/accessory/controller/ [Accessed 4 Aug 2021].

HTC, 2017. *VIVE Tracker | VIVE United Kingdom* [online]. Available from: https://www.vive.com/uk/accessory/vive-tracker/ [Accessed 25 May 2021].

Huhti, J. P., 2019. The Effect of High Monitor Refresh Rate on Game Experience. [online]. Available from: https://osuva.uwasa.fi/handle/10024/10196 [Accessed 11 Aug 2021].

Hwang, I., Son, H. and Kim, J. R., 2017. AirPiano: Enhancing music playing experience in virtual reality with mid-air haptic feedback. *In*: *2017 IEEE World Haptics Conference (WHC)*. Presented at the 2017 IEEE World Haptics Conference (WHC), 213–218.

IBM, 2021. *SPSS Statistics - Overview* [online]. Available from: https://www.ibm.com/uk-en/products/spss-statistics [Accessed 18 Jun 2021].

IJsselsteijn, W. A., de Kort, Y. A. W. and Haans, A., 2006. Is This My Hand I See Before Me? The Rubber Hand Illusion in Reality, Virtual Reality, and Mixed Reality. *Presence: Teleoperators and Virtual Environments*, 15 (4), 455–464.

Ito, M., Wakuda, D., Inoue, S., Makino, Y. and Shinoda, H., 2016. High Spatial Resolution Midair Tactile Display Using 70 kHz Ultrasound. *In*: Bello, F., Kajimoto, H., and Visell, Y., eds. *Haptics: Perception, Devices, Control, and Applications*. Cham: Springer International Publishing, 57–67.

Iwamoto, T., Tatezono, M., Hoshi, T. and Shinoda, H., 2008. Airborne ultrasound tactile display. *In*: *ACM SIGGRAPH 2008 new tech demos* [online]. New York, NY, USA: Association for Computing Machinery, 1. Available from: https://doi.org/10.1145/1401615.1401616 [Accessed 25 Aug 2021].

Jackson, B., Jelke, B. and Brown, G., 2018. Yea Big, Yea High: A 3D User Interface for Surface Selection by Progressive Refinement in Virtual Environments. *In*: *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* [online]. Presented at the 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Tuebingen/Reutlingen, Germany: IEEE, 320–326. Available from: https://ieeexplore.ieee.org/document/8447559/ [Accessed 29 Jan 2019].

Jacobs, J. and Froehlich, B., 2011. A soft hand model for physically-based manipulation of virtual objects. *In*: *2011 IEEE Virtual Reality Conference*. Presented at the 2011 IEEE Virtual Reality Conference, 11–18.

Jankowski, J. and Hachet, M., 2015. Advances in Interaction with 3D Environments. *Computer Graphics Forum*, 34 (1), 152–190.

Jupyter, 2021. *Project Jupyter* [online]. Available from: https://www.jupyter.org [Accessed 18 Jun 2021].

Kamel Boulos, M. N., Blanchard, B. J., Walker, C., Montero, J., Tripathy, A. and Gutierrez-Osuna, R., 2011. Web GIS in practice X: a Microsoft Kinect natural user interface for Google Earth navigation. *International Journal of Health Geographics*, 10 (1), 1–14.

Kappus, B. and Long, B., 2018. Spatiotemporal modulation for mid-air haptic feedback from an ultrasonic phased array. *The Journal of the Acoustical Society of America*, 143 (3), 1836–1836.

Kharoub, H., Lataifeh, M. and Ahmed, N., 2019. 3D User Interface Design and Usability for Immersive VR. *Applied Sciences*, 9 (22), 4861.

Khundam, C., 2015. First person movement control with palm normal and hand gesture interaction in virtual reality. *In*: *2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. Presented at the 2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE), 325–330.

Kid Sense, 2011. Fine Motor Development Checklist. *Kid Sense Child Development* [online]. Available from: https://childdevelopment.com.au/resources/child-development-charts/fine-motor-developmental-checklist/ [Accessed 26 May 2021].

Kim, H. and Choi, Y., 2019a. Performance Comparison of User Interface Devices for Controlling Mining Software in Virtual Reality Environments. *Applied Sciences*, 9 (13), 2584.

Kim, H. and Choi, Y., 2019b. Performance Comparison of User Interface Devices for Controlling Mining Software in Virtual Reality Environments. *Applied Sciences*, 9 (13), 2584.

Kim, H. G., Baddar, W. J., Lim, H., Jeong, H. and Ro, Y. M., 2017. Measurement of exceptional motion in VR video contents for VR sickness assessment using deep convolutional autoencoder. *In*: *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology* [online]. Gothenburg, Sweden: Association for Computing Machinery, 1–7. Available from: https://doi.org/10.1145/3139131.3139137 [Accessed 28 Apr 2020].

Kim, M., Jeon, C. and Kim, J., 2017. A Study on Immersion and Presence of a Portable Hand Haptic System for Immersive Virtual Reality. *Sensors*, 17 (5), 1141.

Koskinen, E., Kaaresoja, T. and Laitinen, P., 2008. Feel-good touch: finding the most pleasant tactile feedback for a mobile touch screen button. *In*: *Proceedings of the 10th international conference on Multimodal interfaces* [online]. New York, NY, USA: Association for Computing Machinery, 297–304. Available from: https://doi.org/10.1145/1452392.1452453 [Accessed 27 Aug 2021].

Kreimeier, J., Hammer, S., Friedmann, D., Karg, P., Bühner, C., Bankel, L. and Götzelmann, T., 2019. Evaluation of different types of haptic feedback influencing the task-based presence and performance in virtual reality. *In*: *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments* [online]. New York, NY, USA: Association for Computing Machinery, 289–298. Available from: https://doi.org/10.1145/3316782.3321536 [Accessed 26 Aug 2021].

LaViola, J. J., 2000. A discussion of cybersickness in virtual environments. *ACM SIGCHI Bulletin*, 32 (1), 47–56.

LaViola, J. J., Kruijff, E., McMahan, R. P., Bowman, D. A. and Poupyrev, I., 2017. *3D user interfaces: theory and practice*. Second edition. Boston: Addison-Wesley.

Leap Motion, 2012. *Leap Motion Technology* [online]. Leap Motion. Available from: https://www.leapmotion.com/technology/ [Accessed 30 Sep 2019].

Leap Motion, 2013. *Tracking | Leap Motion Controller* [online]. Available from: https://www.ultraleap.com/product/leap-motion-controller/ [Accessed 28 Apr 2020].

Leap Motion, 2017. *Unity Modules: Interaction Engine* [online]. Available from: https://leapmotion.github.io/UnityModules/interaction-engine.html [Accessed 4 Sep 2021].

Levac, D. E., Huber, M. E. and Sternad, D., 2019. Learning and transfer of complex motor skills in virtual reality: a perspective review. *Journal of NeuroEngineering and Rehabilitation*, 16 (1), 121.

Li, R., Liu, Z. and Tan, J., 2019. A survey on 3D hand pose estimation: Cameras, methods, and datasets. *Pattern Recognition*, 93, 251–272.

Lin, J., Guo, X., Shao, J., Jiang, C., Zhu, Y. and Zhu, S.-C., 2016. A virtual reality platform for dynamic human-scene interaction. *In*: *SIGGRAPH ASIA 2016 Virtual Reality meets Physical Reality: Modelling and Simulating Virtual Humans and Environments* [online]. New York, NY, USA: Association for Computing Machinery, 1–4. Available from: https://doi.org/10.1145/2992138.2992144 [Accessed 27 Aug 2021].

Linkenauger, S. A., Leyrer, M., Bülthoff, H. H. and Mohler, B. J., 2013. Welcome to Wonderland: The Influence of the Size and Shape of a Virtual Hand On the Perceived Size and Shape of Virtual Objects. *PLOS ONE*, 8 (7), e68594.

Liu, H., Zhang, Z., Xie, X., Zhu, Y., Liu, Y., Wang, Y. and Zhu, S.-C., 2019. High-Fidelity Grasping in Virtual Reality using a Glove-based System. *In*: *2019 International Conference on Robotics and Automation (ICRA)*. Presented at the 2019 International Conference on Robotics and Automation (ICRA), 5180–5186.

Llorach, G., Evans, A. and Blat, J., 2014. Simulator sickness and presence using HMDs: comparing use of a game controller and a position estimation system. *In*: *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology* [online]. New York, NY, USA: Association for Computing Machinery, 137–140. Available from: https://doi.org/10.1145/2671015.2671120 [Accessed 27 May 2021].

Lu, S., Metaxas, D., Samaras, D. and Oliensis, J., 2003. Using multiple cues for hand tracking and model refinement. *In*: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* Presented at the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings., II–443.

Lynx, 2021. *Lynx: The Future of Mixed Reality, now.* [online]. Available from: https://lynx-r.com/ [Accessed 9 Sep 2021].

Ma, Z., Edge, D., Findlater, L. and Tan, H. Z., 2015. Haptic keyclick feedback improves typing speed and reduces typing errors on a flat keyboard. *In*: *2015 IEEE World Haptics Conference (WHC)*. Presented at the 2015 IEEE World Haptics Conference (WHC), 220–227.

Maggioni, C., 1993. A novel gestural input device for virtual reality. *In*: *Proceedings of IEEE Virtual Reality Annual International Symposium*. Presented at the Proceedings of IEEE Virtual Reality Annual International Symposium, 118–124.

Malizia, A. and Bellucci, A., 2012. The artificiality of natural user interfaces. *Communications of the ACM*, 55 (3), 36–38.

Martinez, J., Griffiths, D., Biscione, V., Georgiou, O. and Carter, T., 2018. Touchless Haptic Feedback for Supernatural VR Experiences. *In*: *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. Presented at the 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 629–630.

Martinez, J., Harwood, A., Limerick, H., Clark, R. and Georgiou, O., 2019. Mid-Air Haptic Algorithms for Rendering 3D Shapes. *In*: *2019 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*. Presented at the 2019

IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE), 1–6.

Masurovsky, A., Chojecki, P., Runde, D., Lafci, M., Przewozny, D. and Gaebler, M., 2020. Controller-Free Hand Tracking for Grab-and-Place Tasks in Immersive Virtual Reality: Design Elements and Their Empirical Study. *Multimodal Technologies and Interaction*, 4 (4), 91.

Mayor, J., Raya, L. and Sanchez, A., 2019. A comparative study of virtual reality methods of interaction and locomotion based on presence, cybersickness and usability. *IEEE Transactions on Emerging Topics in Computing*, 1–1.

Mcgurk, H. and Macdonald, J., 1976. Hearing lips and seeing voices. *Nature*, 264 (5588), 746–748.

McKenzie, C. and Glazier, A., 2017. Designing Screen Interfaces for VR. *Google I/O* [online], 2017. Available from: https://www.youtube.com/watch?v=ES9jArHRFHQ.

van der Meijden, O. A. J. and Schijven, M. P., 2009. The value of haptic feedback in conventional and robot-assisted minimal invasive surgery and virtual reality training: a current review. *Surgical Endoscopy*, 23 (6), 1180–1190.

Mendes, D., Caputo, F. M., Giachetti, A., Ferreira, A. and Jorge, J., 2019. A Survey on 3D Virtual Object Manipulation: From the Desktop to Immersive Virtual Environments. *Computer Graphics Forum*, 38 (1), 21–45.

Microsoft, 2001. Xbox controller. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=Xbox_controller&oldid=1035492549 [Accessed 4 Aug 2021].

Microsoft, 2010. Kinect. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=Kinect&oldid=1034468707 [Accessed 5 Aug 2021].

Microsoft, 2016. Microsoft HoloLens. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=Microsoft_HoloLens&oldid=1037342160 [Accessed 19 Aug 2021].

Microsoft, 2019a. *Azure Kinect DK – Develop AI models | Microsoft Azure* [online]. Available from: https://azure.microsoft.com/en-gb/services/kinect-dk/ [Accessed 28 Apr 2020].

Microsoft, 2019b. *MRTK-Unity Developer Documentation - Mixed Reality Toolkit* [online]. Available from: https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/ [Accessed 27 May 2021].

Moehring, M. and Froehlich, B., 2011. Effective manipulation of virtual objects within arm's reach. *In*: *2011 IEEE Virtual Reality Conference*. Presented at the 2011 IEEE Virtual Reality Conference, 131–138.

Moro, C., Štromberga, Z. and Stirling, A., 2017. Virtualisation devices for student learning: Comparison between desktop-based (Oculus Rift) and mobile-based (Gear VR) virtual reality in medical and health science education. *Australasian Journal of Educational Technology* [online], 33 (6). Available from: https://ajet.org.au/index.php/AJET/article/view/3840 [Accessed 27 May 2021].

Motola-Barnes, M., 2018. *Haptic Actuators: Comparing Piezo to ERM and LRA* [online]. Available from: https://blog.piezo.com/haptic-actuators-comparing-piezo-erm-lra [Accessed 23 Aug 2021].

NASA, 1986. *TLX @ NASA Ames - Home* [online]. Available from: https://humansystems.arc.nasa.gov/groups/TLX/ [Accessed 27 May 2021].

Nintendo, 1997. Rumble Pak. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=Rumble_Pak&oldid=1035331583 [Accessed 4 Aug 2021].

Nintendo, 2006. Wii. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=Wii&oldid=1039663481 [Accessed 5 Sep 2021].

Nintendo, 2017. *Nintendo Switch - Wikipedia* [online]. Available from: https://en.wikipedia.org/wiki/Nintendo_Switch [Accessed 23 Aug 2021].

Nirei, K., Saito, H., Mochimaru, M. and Ozawa, S., 1996. Human hand tracking from binocular image sequences. *In*: *22nd International Conference on Industrial Electronics, Control, and Instrumentation Proceedings of the 1996 IEEE IECON*. Presented at the 22nd International Conference on Industrial Electronics, Control, and Instrumentation Proceedings of the 1996 IEEE IECON, 297–302 vol.1.

Nor'a, M. N. A. and Ismail, A. W., 2019. Integrating Virtual Reality and Augmented Reality in a Collaborative User Interface. *International Journal of Innovative Computing* [online], 9 (2). Available from: https://ijic.utm.my/index.php/ijic/article/view/242 [Accessed 21 Aug 2021].

Nordvall, M., 2014. The Sightlence Game: Designing a Haptic Computer Game Interface. [online]. Available from: http://www.digra.org/wp-content/uploads/digital-library/paper_473.pdf.

Novint, 2006. Novint Technologies. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=Novint_Technologies&oldid=1040009515 [Accessed 24 Aug 2021].

Nukarinen, T., Kangas, J., Rantala, J., Koskinen, O. and Raisamo, R., 2018. Evaluating ray casting and two gaze-based pointing techniques for object selection in virtual reality. *In*: *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology* [online]. New York, NY, USA: Association for Computing Machinery, 1–2. Available from: https://doi.org/10.1145/3281505.3283382 [Accessed 5 Sep 2021].

OBS, 2021. *Open Broadcaster Software | OBS* [online]. Available from: https://obsproject.com/ [Accessed 17 Jun 2021].

Oculus, 2016. *Oculus Rift | Oculus* [online]. Available from: https://www.oculus.com/rift/ [Accessed 25 May 2021].

Oculus, 2019a. *Oculus Quest: All-in-one VR Headset | Oculus* [online]. Available from: https://www.oculus.com/quest/ [Accessed 28 Apr 2020].

Oculus, 2019b. *Oculus Quest: All-in-one VR Headset | Oculus* [online]. Available from: https://www.oculus.com/quest/ [Accessed 28 Apr 2020].

Ogris, G., Stiefmeier, T., Junker, H., Lukowicz, P. and Troster, G., 2005. Using ultrasonic hand tracking to augment motion analysis based recognition of manipulative gestures. *In*: *Ninth IEEE International Symposium on Wearable Computers (ISWC'05)*. Presented at the Ninth IEEE International Symposium on Wearable Computers (ISWC'05), 152–159.

OpenCV, 2021. *OpenCV* [online]. OpenCV. Available from: https://opencv.org/ [Accessed 17 Jun 2021].

Oqua, O. I., Chen, J., Annamalai, A. and Yang, C., 2018. 3D Printed Data Glove Design for VR Based Hand Gesture Recognition. *In*: *2018 11th International Workshop on Human Friendly Robotics (HFR)*. Presented at the 2018 11th International Workshop on Human Friendly Robotics (HFR), 66–71.

Pan, Z., Li, Y., Zhang, M., Sun, C., Guo, K., Tang, X. and Zhou, S. Z., 2010. A real-time multi-cue hand tracking algorithm based on computer vision. *In*: *2010 IEEE Virtual Reality Conference (VR)*. Presented at the 2010 IEEE Virtual Reality Conference (VR), 219–222.

pandas, 2021. *pandas - Python Data Analysis Library* [online]. Available from: https://pandas.pydata.org/ [Accessed 18 Jun 2021].

Panteleris, P., Oikonomidis, I. and Argyros, A., 2018. Using a Single RGB Frame for Real Time 3D Hand Pose Estimation in the Wild. *In*: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Presented at the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 436–445.

Park, G., Choi, S., Hwang, K., Kim, S., Sa, J. and Joung, M., 2011. Tactile effect design and evaluation for virtual buttons on a mobile device touchscreen. *In*: *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services* [online]. New York, NY, USA: Association for Computing Machinery, 11–20. Available from: https://doi.org/10.1145/2037373.2037376 [Accessed 27 Aug 2021].

Park, J., Han, I. and Lee, W., 2019. Effect of Haptic Feedback on the Perceived Size of a Virtual Object. *IEEE Access*, 7, 83673–83681.

Park, K., Kim, D. and Han, S. H., 2020. Usability of the size, spacing, and operation method of virtual buttons with virtual hand on head-mounted displays. *International Journal of Industrial Ergonomics*, 76, 102939.

Peeva, D., Baird, B., Izmirli, O. and Blevins, D., 2004. Haptic and sound correlations: pitch, loudness and texture. *In*: *Proceedings. Eighth International Conference on Information Visualisation, 2004. IV 2004.* Presented at the Proceedings. Eighth International Conference on Information Visualisation, 2004. IV 2004., 659–664.

Perry, A., 2020. *A 120Hz smartphone display is nice to have, but do you need it?* [online]. Mashable. Available from: https://mashable.com/article/refresh-rate-explained-120hz-smartphones/ [Accessed 29 Apr 2020].

Pezent, E., Israr, A., Samad, M., Robinson, S., Agarwal, P., Benko, H. and Colonnese, N., 2019. Tasbi: Multisensory Squeeze and Vibrotactile Wrist Haptics for Augmented and Virtual Reality. *In*: *2019 IEEE World Haptics Conference (WHC)*. Presented at the 2019 IEEE World Haptics Conference (WHC), 1–6.

Pfeuffer, K., Mayer, B., Mardanbegi, D. and Gellersen, H., 2017. Gaze + pinch interaction in virtual reality. *In*: *Proceedings of the 5th Symposium on Spatial User Interaction* [online]. New York, NY, USA: Association for Computing Machinery, 99–108. Available from: https://doi.org/10.1145/3131277.3132180 [Accessed 5 Sep 2021].

Pham, D.-M. and Stuerzlinger, W., 2019. Is the Pen Mightier than the Controller? A Comparison of Input Devices for Selection in Virtual and Augmented Reality. *In*: *25th ACM Symposium on Virtual Reality Software and Technology* [online]. New York, NY, USA: Association for Computing Machinery, 1–11. Available from: https://doi.org/10.1145/3359996.3364264 [Accessed 27 Aug 2021].

Pimax, 2021. *PiMAX - PiMax* [online]. Available from: https://pimax.com/ [Accessed 9 Sep 2021].

Poupyrev, I., Billinghurst, M., Weghorst, S. and Ichikawa, T., 1996. The Go-Go Interaction Technique: Non-linear Mapping for Direct Manipulation in VR, 2.

Prachyabrued, M. and Borst, C. W., 2012. Virtual grasp release method and evaluation. *International Journal of Human-Computer Studies*, 70 (11), 828–848.

Precision, 2015. AB-004 : Understanding ERM Vibration Motor Characteristics - Precision Microdrives. [online]. Available from: https://www.precisionmicrodrives.com/content/ab-004-understanding-erm-vibration-motor-characteristics/ [Accessed 23 Aug 2021].

Price, A. and Long, B., 2018. Fibonacci Spiral Arranged Ultrasound Phased Array for Mid-Air Haptics. *In*: *2018 IEEE International Ultrasonics Symposium (IUS)*. Presented at the 2018 IEEE International Ultrasonics Symposium (IUS), 1–4.

Python, 2021. *Welcome to Python.org* [online]. Python.org. Available from: https://www.python.org/ [Accessed 18 Jun 2021].

R, 2021. *R: The R Project for Statistical Computing* [online]. Available from: https://www.r-project.org/ [Accessed 18 Jun 2021].

Rehg, J. M. and Kanade, T., 1994. Visual tracking of high DOF articulated structures: An application to human hand tracking. *In*: Eklundh, J.-O., ed. *Computer Vision — ECCV '94*. Berlin, Heidelberg: Springer, 35–46.

Reuter, E.-M., Voelcker-Rehage, C., Vieluf, S. and Godde, B., 2012. Touch perception throughout working life: effects of age and expertise. *Experimental Brain Research*, 216 (2), 287–297.

Riecke, B. E., LaViola, J. J. and Kruijff, E., 2018. 3D user interfaces for virtual reality and games: 3D selection, manipulation, and spatial navigation. *In*: *ACM SIGGRAPH 2018 Courses on - SIGGRAPH '18* [online]. Presented at the ACM SIGGRAPH 2018 Courses, Vancouver, British Columbia, Canada: ACM Press, 1–94. Available from: http://dl.acm.org/citation.cfm?doid=3214834.3214869 [Accessed 29 Jan 2019].

Robertson, G. G., Henderson, D. A. and Card, S. K., 1991. Buttons as first class objects on an X desktop. *In*: *Proceedings of the 4th annual ACM symposium on User interface software and technology* [online]. New York, NY, USA: Association for Computing Machinery, 35–44. Available from: https://doi.org/10.1145/120782.120786 [Accessed 20 Aug 2021].

Rubin, P., 2014. The Inside Story of Oculus Rift and How Virtual Reality Became Reality. *Wired* [online]. Available from: https://www.wired.com/2014/05/oculus-rift-4/ [Accessed 13 Sep 2021].

Rutten, I. and Geerts, D., 2020. Better Because It's New: The Impact of Perceived Novelty on the Added Value of Mid-Air Haptic Feedback. *In*: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* [online]. New York, NY, USA: Association for Computing Machinery, 1–13. Available from: https://doi.org/10.1145/3313831.3376668 [Accessed 11 Sep 2021].

Ryan, M.-L., 2015. *Narrative as virtual reality 2: Revisiting immersion and interactivity in literature and electronic media*. JHU Press.

Sagayam, K. M. and Hemanth, D. J., 2017. Hand posture and gesture recognition techniques for virtual reality applications: a survey. *Virtual Reality*, 21 (2), 91–107.

Samad, M., Gatti, E., Hermes, A., Benko, H. and Parise, C., 2019. Pseudo-Haptic Weight: Changing the Perceived Weight of Virtual Objects By Manipulating Control-Display Ratio. *In*: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* [online]. Presented at the CHI '19: CHI Conference on Human Factors in Computing Systems, Glasgow Scotland Uk: ACM, 1–13. Available from: https://dl.acm.org/doi/10.1145/3290605.3300550 [Accessed 6 Sep 2021].

Sato, Y., Kobayashi, Y. and Koike, H., 2000. Fast tracking of hands and fingertips in infrared images for augmented desk interface. *In*: *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*. Presented at the Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), 462–467.

Sensable Technologies, 1994. "Phantom Omni" – 6 DOF master device | Delft Haptics Lab. [online]. Available from: https://delfthapticslab.nl/device/phantom-omni/ [Accessed 26 May 2021].

SenseGlove, 2019. *SenseGlove | Make the Digital Feel Real* [online]. SenseGlove. Available from: https://www.senseglove.com/ [Accessed 26 Aug 2021].

Sharex, 2021. *ShareX - The best free and open source screenshot tool for Windows* [online]. Available from: https://getsharex.com/ [Accessed 17 Jun 2021].

Silva, E. S., Teichrieb, V. and Ramalho, G. L., 2013. A Preliminary Evaluation of the Leap Motion Sensor as Controller of New Digital Musical Instruments, 12.

Sirenix, 2021. *TeamSirenix/odin-serializer* [online]. TeamSirenix. Available from: https://github.com/TeamSirenix/odin-serializer [Accessed 4 Aug 2021].

Smeragliuolo, A. H., Hill, N. J., Disla, L. and Putrino, D., 2016. Validation of the Leap Motion Controller using markered motion capture technology. *Journal of Biomechanics*, 49 (9), 1742–1750.

Smith, B., Wu, C., Wen, H., Peluse, P., Sheikh, Y., Hodgins, J. K. and Shiratori, T., 2020. Constraining dense hand surface tracking with elasticity. *ACM Transactions on Graphics*, 39 (6), 219:1-219:14.

Son, B. and Park, J., 2018. Haptic Feedback to the Palm and Fingers for Improved Tactile Perception of Large Objects. *In*: *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* [online]. New York, NY, USA: Association for Computing Machinery, 757–763. Available from: https://doi.org/10.1145/3242587.3242656 [Accessed 26 Aug 2021].

Sony, 1997. DualShock. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=DualShock&oldid=1037110018 [Accessed 4 Aug 2021].

Sony, 2003. EyeToy. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=EyeToy&oldid=1039732842 [Accessed 5 Sep 2021].

Sony, 2010. PlayStation Move. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=PlayStation_Move&oldid=1041082755 [Accessed 5 Sep 2021].

Sony, 2016. PlayStation VR. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=PlayStation_VR&oldid=1037546154 [Accessed 5 Sep 2021].

Sony, 2020. PlayStation 5. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=PlayStation_5&oldid=1038186143 [Accessed 24 Aug 2021].

Stoakley, R., Conway, M. J. and Pausch, R., 1995a. Virtual reality on a WIM: interactive worlds in miniature. *In*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* [online]. Denver, Colorado, USA: ACM Press/Addison-Wesley Publishing Co., 265–272. Available from: https://doi.org/10.1145/223904.223938 [Accessed 28 Apr 2020].

Stoakley, R., Conway, M. J. and Pausch, R., 1995b. Virtual reality on a WIM: interactive worlds in miniature. *In*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* [online]. Denver, Colorado, USA: ACM Press/Addison-Wesley Publishing Co., 265–272. Available from: https://doi.org/10.1145/223904.223938 [Accessed 4 May 2020].

Taima, Y., Ban, Y., Narumi, T., Tanikawa, T. and Hirose, M., 2014. Controlling fatigue while lifting objects using Pseudo-haptics in a mixed reality space. *In*: *2014 IEEE Haptics Symposium (HAPTICS)*. Presented at the 2014 IEEE Haptics Symposium (HAPTICS), 175–180.

Talvas, A., Marchal, M. and Lécuyer, A., 2013. The god-finger method for improving 3D interaction with virtual objects through simulation of contact area. *In*: *2013 IEEE Symposium on 3D User Interfaces (3DUI)*. Presented at the 2013 IEEE Symposium on 3D User Interfaces (3DUI), 111–114.

Teather, R. J., Natapov, D. and Jenkin, M., 2010. Evaluating haptic feedback in virtual environments using ISO 9241–9. *In*: *2010 IEEE Virtual Reality Conference (VR)*. Presented at the 2010 IEEE Virtual Reality Conference (VR), 307–308.

TensorFlow, 2021. *TensorFlow* [online]. TensorFlow. Available from: https://www.tensorflow.org/ [Accessed 17 Jun 2021].

Thompson, S., 2016. *Virtual Reality and the Resolution of the Human Eye* [online]. Medium. Available from: https://medium.com/@FreneticPony/virtual-reality-and-the-resolution-of-the-human-eye-5e601b0ef030 [Accessed 28 Apr 2020].

Tung, J. Y., Lulic, T., Gonzalez, D. A., Tran, J., Dickerson, C. R. and Roy, E. A., 2015. Evaluation of a portable markerless finger position capture device: accuracy of the Leap Motion controller in healthy adults. *Physiological Measurement*, 36 (5), 1025–1035.

Ultrahaptics, 2017. *Ultrahaptics' Mid-air touch with ultrasound - how it works - YouTube* [online]. Available from: https://www.youtube.com/watch?v=tBMkBS_Rlgc [Accessed 29 Apr 2020].

Ultrahaptics, 2018. *Ultrahaptics Stratos Explore* [online]. Ultrahaptics. Available from: https://www.ultrahaptics.com/products-programs/stratos-explore-development-kit/ [Accessed 30 Sep 2019].

Ultraleap, 2013. *Tracking | Leap Motion Controller | Ultraleap* [online]. Available from: https://www.ultraleap.com/product/leap-motion-controller/ [Accessed 28 Apr 2020].

Ultraleap, 2021a. *Unity Modules: Interaction Engine* [online]. Available from: https://leapmotion.github.io/UnityModules/interaction-engine.html [Accessed 27 May 2021].

Ultraleap, 2021b. *Release Unity Modules 4.5.1 · leapmotion/UnityModules* [online]. GitHub. Available from: /leapmotion/UnityModules/releases/tag/UM-4.5.1 [Accessed 27 May 2021].

Unity, 2021a. *Unity Recorder User Manual | Package Manager UI website* [online]. Available from: https://docs.unity3d.com/Packages/com.unity.recorder@2.0/manual/index.html [Accessed 17 Jun 2021].

Unity, 2021b. *Leverage your game usage data with Unity Analytics | Features | Unity* [online]. Available from: https://unity.com/features/analytics [Accessed 22 Aug 2021].

Unity, 2021c. *Unity Real-Time Development Platform | 3D, 2D VR & AR Engine* [online]. Available from: https://unity.com/ [Accessed 17 Jun 2021].

Unity, 2021d. *Unity - Manual: Order of execution for event functions* [online]. Available from: https://docs.unity3d.com/Manual/ExecutionOrder.html [Accessed 4 Jun 2021].

Valentini, P. P. and Pezzuti, E., 2017. Accuracy in fingertip tracking using Leap Motion Controller for interactive virtual applications. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 11 (3), 641–650.

Valve, 2018. *Skeleton Input | SteamVR Unity Plugin* [online]. Available from: https://valvesoftware.github.io/steamvr_unity_plugin/articles/Skeleton-Input.html [Accessed 27 May 2021].

Valve, 2019a. *Valve Index* [online]. Valve Corporation. Available from: https://www.valvesoftware.com/en/index [Accessed 5 Aug 2021].

Valve, 2019b. *Valve Index Controllers* [online]. Valve Corporation. Available from: https://www.valvesoftware.com/en/index/controllers [Accessed 25 May 2021].

Våpenstad, C., Hofstad, E. F., Bø, L. E., Kuhry, E., Johnsen, G., Mårvik, R., Langø, T. and Hernes, T. N., 2017. Lack of transfer of skills after virtual reality simulator training with haptic feedback. *Minimally Invasive Therapy & Allied Technologies*, 26 (6), 346–354.

Våpenstad, C., Hofstad, E. F., Langø, T., Mårvik, R. and Chmarra, M. K., 2013. Perceiving haptic feedback in virtual reality simulators. *Surgical Endoscopy*, 27 (7), 2391–2397.

Varga, S., 2021. *Haptic gloves for virtual reality and robotics* [online]. HaptX. Available from: https://haptx.com/ [Accessed 25 May 2021].

Varjo, 2019. *Varjo VR-2 Pro - Varjo.com* [online]. Available from: https://varjo.com/products/vr-2-pro/ [Accessed 28 Apr 2020].

Vélaz, Y., Rodríguez Arce, J., Gutiérrez, T., Lozano-Rodero, A. and Suescun, A., 2014. The Influence of Interaction Technology on the Learning of Assembly Tasks Using Virtual Reality. *Journal of Computing and Information Science in Engineering* [online], 14 (4). Available from: https://doi.org/10.1115/1.4028588 [Accessed 7 Sep 2021].

Visconti, P., Gaetani, F., Zappatore, G. A. and Primiceri, P., 2018. Technical Features and Functionalities of Myo Armband: An Overview on Related Literature and Advanced Applications of Myoelectric Armbands Mainly Focused on Arm Prostheses. *International Journal on Smart Sensing and Intelligent Systems* [online], 11 (1). Available from: https://doi.org/10.21307/ijssis-2018-005 [Accessed 31 Aug 2021].

Webb, M., Tracey, M., Harwin, W., Tokatli, O., Hwang, F., Johnson, R., Barrett, N. and Jones, C., 2022. Haptic-enabled collaborative learning in virtual reality for schools. *Education and Information Technologies*, 27 (1), 937–960.

Weiß, Y., Hepperle, D., Sieß, A. and Wölfel, M., 2018. What User Interface to Use for Virtual Reality? 2D, 3D or Speech–A User Study. *In*: *2018 International Conference on Cyberworlds (CW)*. Presented at the 2018 International Conference on Cyberworlds (CW), 50–57.

Wikipedia, 2011. Razer Hydra. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=Razer_Hydra&oldid=996767841 [Accessed 26 May 2021].

Wikipedia, 2021a. Room scale. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=Room_scale&oldid=1003744675 [Accessed 25 Aug 2021].

Wikipedia, 2021b. Ferrofluid. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=Ferrofluid&oldid=1024065043 [Accessed 5 Aug 2021].

Wikipedia, 2021c. Oculus Go. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=Oculus_Go&oldid=1023550172 [Accessed 25 May 2021].

Wikipedia, 2021d. Fine motor skill. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=Fine_motor_skill&oldid=1000525074 [Accessed 26 May 2021].

Wikipedia, 2021e. Uncompressed video. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=Uncompressed_video&oldid=1018518090 [Accessed 17 Jun 2021].

Wikipedia, 2021f. Open–closed principle. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=Open%E2%80%93closed_principle&oldid=1024323953 [Accessed 4 Aug 2021].

Wikipedia, 2021. *Thread pool - Wikipedia* [online]. Available from: https://en.wikipedia.org/wiki/Thread_pool [Accessed 1 Jun 2021].

Wikipedia, 2021g. Observer pattern. *Wikipedia* [online]. Available from: https://en.wikipedia.org/w/index.php?title=Observer_pattern&oldid=1027272229 [Accessed 3 Aug 2021].

Wilson, G., Carter, T., Subramanian, S. and Brewster, S. A., 2014. Perception of ultrasonic haptic feedback on the hand: localisation and apparent motion. *In*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* [online]. Toronto, Ontario, Canada: Association for Computing Machinery, 1133–1142. Available from: https://doi.org/10.1145/2556288.2557033 [Accessed 28 Apr 2020].

Wingrave, C. and Bowman, D., 2005. Baseline factors for raycasting selection. *In*: *Proceedings of HCI International*. Citeseer, 61–68.

Wu, C.-M., Hsu, C.-W., Lee, T.-K. and Smith, S., 2017. A virtual reality keyboard with realistic haptic feedback in a fully immersive virtual environment. *Virtual Reality*, 21 (1), 19–29.

Young, M. K., Gaylor, G. B., Andrus, S. M. and Bodenheimer, B., 2014. A comparison of two cost-differentiated virtual reality systems for perception and action tasks. *In*: *Proceedings of the ACM Symposium on Applied Perception* [online]. New York, NY, USA: Association for Computing Machinery, 83–90. Available from: https://doi.org/10.1145/2628257.2628261 [Accessed 27 May 2021].

YouTube, 2021. *System requirements - YouTube Help* [online]. Available from: https://support.google.com/youtube/answer/78358?hl=en-GB [Accessed 17 Jun 2021].

Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L. and Grundmann, M., 2020. MediaPipe Hands: On-device Real-time Hand Tracking. *arXiv:2006.10214 [cs]* [online]. Available from: http://arxiv.org/abs/2006.10214 [Accessed 22 Jun 2021].

Zhou, M., Tse, S., Derevianko, A., Jones, D. B., Schwaitzberg, S. D. and Cao, C. G. L., 2012. Effect of haptic feedback in laparoscopic surgery skill acquisition. *Surgical Endoscopy*, 26 (4), 1128–1134.

Zimmerman, T. G., Lanier, J., Blanchard, C., Bryson, S. and Harvill, Y., 1986. A hand gesture interface device. *ACM SIGCHI Bulletin*, 17 (SI), 189–192.