



Article

A Comparison of One- and Two-Handed Gesture User Interfaces in Virtual Reality—A Task-Based Approach

Taneli Nyysönen * , Seppo Helle, Teijo Lehtonen and Jouni Smed

Department of Computing, University of Turku, 20014 Turku, Finland; seilhe@utu.fi (S.H.); tetale@utu.fi (T.L.); jouni.smed@utu.fi (J.S.)

* Correspondence: ttjnyy@utu.fi

Abstract: This paper presents two gesture-based user interfaces which were designed for a 3D design review in virtual reality (VR) with inspiration drawn from the shipbuilding industry's need to streamline and make their processes more sustainable. The user interfaces, one focusing on single-hand (unimanual) gestures and the other focusing on dual-handed (bimanual) usage, are tested as a case study using 13 tasks. The unimanual approach attempts to provide a higher degree of flexibility, while the bimanual approach seeks to provide more control over the interaction. The interfaces were developed for the Meta Quest 2 VR headset using the Unity game engine. Hand-tracking (HT) is utilized due to potential usability benefits in comparison to standard controller-based user interfaces, which lack intuitiveness regarding the controls and can cause more strain. The user interfaces were tested with 25 test users, and the results indicate a preference toward the one-handed user interface with little variation in test user categories. Additionally, the testing order, which was counterbalanced, had a statistically significant impact on the preference and performance, indicating that learning novel interaction mechanisms requires an adjustment period for reliable results. VR sickness was also strongly experienced by a few users, and there were no signs that gesture controls would significantly alleviate it.

Keywords: virtual reality; hand-tracking; gesture; user interface; 3D design review; comparison study



Citation: Nyysönen, T.; Helle, S.; Lehtonen, T.; Smed, J. A Comparison of One- and Two-Handed Gesture User Interfaces in Virtual Reality—A Task-Based Approach. *Multimodal Technol. Interact.* **2024**, *8*, 10. <https://doi.org/10.3390/mti8020010>

Academic Editor: Arun K. Kulshreshth

Received: 9 January 2024

Revised: 26 January 2024

Accepted: 28 January 2024

Published: 2 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Efforts have been made over the past several years to integrate virtual reality (VR) into industrial construction-related processes [1], and in the case of our research, particularly for reviewing 3D designs. However, the uptake of VR in these areas has been gradual. This slow adoption can be attributed to various factors such as concerns about reliability, user comfort, continuously evolving technology landscapes, and the complexities of VR systems. Additionally, high costs and the steep learning curve associated with these technologies present significant entry barriers [1,2]. Despite its challenges, our previous results [3] suggest that utilizing hand tracking (HT) seems promising for 3D design review applications in VR. Additionally, there are some data to support the effectiveness of VR usage in design reviews as a performance enhancer [4]. Our next step is to determine what type of gestures (unimanual, bimanual symmetric, or bimanual asymmetric, as described in Section 2) are most effective in these types of use case scenarios and in VR overall, and for that reason, two HT user interfaces (UIs) were designed which attempt to gain more insight on the usage of gesture actions. The interfaces approach the problem of higher VR adoption from two angles: (1) to maximize action initiation speed and efficiency by reducing the complexity of the gesture controls in addition to affording multiple actions to be taken simultaneously (unimanual approach) and (2) to minimise input errors and loss of control by requiring more complicated gestures and only one action at a time (bimanual approach).

There are three main research questions in this paper:

1. Which is preferred when handling and viewing 3D models in virtual environments: fewer input errors and unintentional inputs (the two-handed user interface) or simpler functionality with multitasking options (the one-handed user interface)?
2. Which type of gesture actions (unimanual, bimanual symmetric, or bimanual asymmetric, as described in Section 2) is more usable in a VR 3D design review?
3. What usability issues arise with either user interface, and which approach creates a better user experience overall?

Further impetus for examining the effectiveness of 3D design review software comes from the Sustainable Shipbuilding Concepts (SusCon, [5]) research project. This initiative focuses on developing eco-friendly methods within the shipbuilding sector. A secondary objective of SusCon is to evaluate the use of VR in the ship design process, primarily considering it as a tool for visualizing various components during the design review stage. The project contributes to sustainability by minimizing the need for travel and physical prototypes. Throughout the SusCon project, there has been close collaboration with the shipbuilding industry. Several industry partners have already begun experimenting with VR using controller-based systems for visualization purposes. Some of them also tested HT for the same concept in our previous study [3] with results showing positive reception toward HT in VR when compared to controllers. While the inspiration for the study initially emerged from the shipbuilding industry, the applications for the technology are all-encompassing, and the 3D design review can be utilized in other sectors of industry, education, and entertainment.

Our findings indicate that the unimanual approach and thus the simplicity of controls is preferred over increased action control, which signals that action initiation speed seems to be important when moving and interacting inside virtual environments viewing 3D models. Additionally, there seems to be a strong link between learning a novel interaction mechanic based on the testing order, which was used to counterbalance the study, showing that the preference and performance in either user interface increased considerably if the test users already had experience in the other. This is likely also influenced by the then increased familiarity with the testing setup. Furthermore, there were more issues with learning some of the gesture actions than others: namely, menu usage was difficult for many users, which reflects on the fist recognition and subconscious fist closing or opening by the users.

2. Related Work

Previous HT-related research in VR in general is connected to two areas: (1) creating custom hand-tracking mechanisms and, for example, comparing different tracking algorithms and their accuracy, e.g., [6–8], or (2) evaluating the design of gestures and in what kind of scenarios they can be utilized the best. The focus of our literature review lies in the latter section, as this study is focused on usability design instead of refining or experimenting with novel tracking mechanisms. The gestures are separated into three categories: *mid-air gestures*, *gestures performed with hand-held devices*, and *touch-based gestures* (touchpads on controllers or HMDs) [9]. This study focuses solely on the mid-air gestures; thus, so does the literature review.

There has been a lot of research related to analyzing 3D user interfaces in the past, and the terms *tangible* or *graspable* interface are mentioned frequently, referring to digital interfaces which are interactable via the physical dimension in some way [10–13], e.g., a computer mouse using a personal computer. Some UI-focused studies include object-grabbing with HT (mid-air grabbing) in VR, e.g., [14] or selecting objects from interactive displays and surfaces [15–17]. A lot of the research on gesture usage is empirical (by nature) [12,18] and in some cases, it has been influenced by concepts such as VR animation design [19]. Based on the existing literature, there is ample room for additional research related to usability benefits and issues of gestures and natural tracking. Some major issues which have been identified are strain on body parts (e.g., fingers, hands, arms, or shoulders) and involuntary gesture activation due to imprecise tracking or user hand movements.

There are also some case studies using augmented reality (AR) which have attempted to solve issues in hand interaction, e.g., by combining hands with gaze usage for selection tasks [20] or comparing the differences of standard interaction devices and gestures for rotating and translating 3D models in space [21]. In general, at least for VR, the benefits of hand interaction include, e.g., naturalness, potential ease of learning, and the lack of needing heavy controllers, thus leading to lesser strain [3,16,18,22].

The study's main focus lies in comparing one- and two-handed gesture interaction, which have been compared previously by, e.g., [23–25]. A strongly related concept is the classification of human manual activities by Guiard [23], which attempts to understand how work is distributed between the right and the left hand. There are three main categories: *unimanual*, which requires only one-hand coordination, e.g., throwing certain objects such as a javelin or using dining utensils like a spoon for soup, *bimanual symmetric*, where both hands coordinate with identical action in phase (e.g., weightlifting) or out of phase (e.g., milking a cow), and *bimanual asymmetric*, which requires complex dual-handed coordination, e.g., playing musical instruments [23]. This study contains gestures belonging to all three categories.

Schäfer et al. (2021) [25] compared two one-handed and two-handed UIs in a teleportation-based locomotion system. Although the article only discusses teleporting and not continuous locomotion used in this article, the results suggest that one-handed gestures were deemed faster to initiate by those who favored them, whereas two-handed gestures were perceived as more reliable. The premise of increased reliability for the two-handed gestures and faster initiation speed for the one-handed gestures is agreed upon and was used as a hypothesis for this study.

There is also a need for better specifications of learning evaluation parameters apart from performance (scores, completion time) and usability in relation to training the users to complete tasks in VR [26]. In this case study, the controls are taught to the users in the form of tasks, meaning learning evaluation is an important aspect. Moreover, the underlying idea is that the HT technology, if implemented correctly, can offer improvements also in the area of learning [27–29].

An important part of HT UI design is the gesture design: what gestures to use and why? Gesture use strongly relates to *semiotics*, which refers to the study of signs and symbols and their use or interpretation. Usually, semiotics are used to communicate with another person, and there is plenty of research related to this [30], but it is also important to remember that using gestures even without the intent of communication can potentially invoke (undesired or desired) reactions in others. For example, a gesture can have a specific personal meaning to a person viewing it unbeknownst to the gesture initiator. This is why the designers of HT UIs in VR need to consider at least the following aspects of *pragmatic semiotics* in VR [31], or in other words, interaction:

- Is a given gesture familiar to the target audience from everyday life? If it is, then what kind of actions is it related to? Should this underlying connection be considered when deciding whether to use the gesture or not?
- Does the gesture have some cultural implications to the user (or any user)? Could this affect the efficiency with which the user can complete a certain task using the gesture?
- Could multiple different gestures in combination (e.g., when performed with different hands) result into an undesired, potentially offensive or distracting gesture? How to design around this?

Another issue apart from gesture design with 3D hand-tracking is reliability, especially when utilizing gloveless approaches [32]. Specific hand-tracking gloves, often utilizing accelerometers, gyroscopes and flex sensors, have been invented with great tracking accuracy, but these are often intrusive and expensive (e.g., ShapeHand [33], Hi5 VR glove [34]) or limited in other ways (e.g., a color glove [35,36]). The existing (gloveless) tracking mechanisms are often not sophisticated enough for very precise work because of issues such as occlusion (especially when crossing hands), lighting condition requirements, personal differences in hand size, shape, and default posture, in addition to proneness for external disturbances

(e.g., sunlight for infrared-based tracking systems preventing/hindering outside usage). Thus, while gestures provide increased immersion, they lose in accuracy when compared with the controller-based approach, although there is a potential middle ground in combining the two [22]. For a specific example about HT issues, while the Meta Quest 2 utilized in this study uses mainly visible light for tracking, the headset's lens is very sensitive to the sun's radiation, which is why the device cannot be used outdoors, while the lack of infrared-tracking leaves a vulnerability to lighting conditions even indoors. Additionally, the cameras sometimes misclassifies gestures which appear clear to the human eye for reasons related to the user's immutable traits, namely hand size and posture. These issues are evident in all HT usage but especially in areas where small mistakes can have dire consequences, e.g., construction sites or hospitals, where all potential error factors must be eliminated before the technology adoption should take place. The technology shows promise for the future though with results similar to Wang & Zhu [37] who tested the feasibility toward automated HT signal recognition in the construction industry with high gesture classification accuracy (93.3%).

Another concept that is explored with HT in this study is *interaction fidelity* [38] and how HT could be utilized to design high-fidelity interaction. The interaction fidelity can be analyzed with three metrics: *biomechanical symmetry*, which means how well the used mechanics mirror the simulated real-world actions (e.g., hand motion exactness, haptic feedback), *input veracity*, which refers to the accuracy with which the interaction is captured by the input devices (e.g., latency, camera tracking radius), and *control symmetry*, which describes the degree of exactness of the location where the interaction is taking place (e.g., a hand tracked with HT is near exactly in the same place in VR as in the physical world, whereas a ray going out from the user's hand/controller is not) [39]. The interaction fidelity of virtual interactions, according to McMahan et al. [39], suffers from the *uncanny valley* effect, meaning that medium-fidelity interaction mechanisms are worse in terms of usability than the high (very realistic) and low (not realistic at all). This is why the HT mechanisms should be as high fidelity as possible, which poses design challenges.

3. Hand-Tracking User Interfaces

The original idea behind researching hand-tracking (HT) in VR user interfaces emerged from our attempts to ease the adoption process of VR as a tool, which was mainly to see if HT could improve usability in VR environments compared to a controller-based approach [3]. In this study, a choice was made to design, develop, and compare two HT user interfaces to learn about the challenges and benefits of HT technology more. In essence, the two interfaces, one-handed (1H) and two-handed (2H), aim to find out if there are differences in usability between unimanual and bimanual action initiation and what those potential differences are. Additionally, the effectiveness of the chosen gestures and how they suit either interface is analyzed. Various side factors, e.g., whether HT can lessen VR sickness or reduce physical strain, are also noted down. The potential VR sickness lessening is based on the idea that gesture controls could better represent the self-induced illusory motion, *vection* [40,41], inside VR, and thus reduce the sensory conflicts that stem from it.

Common factors for both interfaces are that the non-pinch gestures need to be continuously recognized for the actions to stay active, while the pinch gestures, index-finger and middle-finger pinch, can only activate once per pinch, and to redo a pinch action, the gesture must first be uninitialized. The latter exists to prevent unwanted chain teleportation or button activation. Additionally, the gestures cannot activate any movement actions while the menu is active to prevent unwanted movement when trying to use the menu and also to give users a chance to rest their arms.

In this test, we wanted to make sure the test users understand when a gesture is being correctly recognized. To achieve this, we introduced a color scheme to highlight when a hand is in one of the three possible states, as shown in Figure 1: (1) a known gesture is being recognized and it activates a corresponding function currently (the hand becomes

green), (2) a known gesture is being recognized, but the rotation of the hand is incorrect, or the menu is currently active; thus, no functionality is activated (the hand becomes yellow), or (3) no known gestures are recognized currently (the hand becomes white). The hands only changed color in cases where a gesture was truly recognized and the actions only fired when the conditions (such as menu being closed) were correct, so there was no possibility for false positives or negatives, e.g., hand showing green and nothing happening or hand showing yellow and some action occurring anyway.

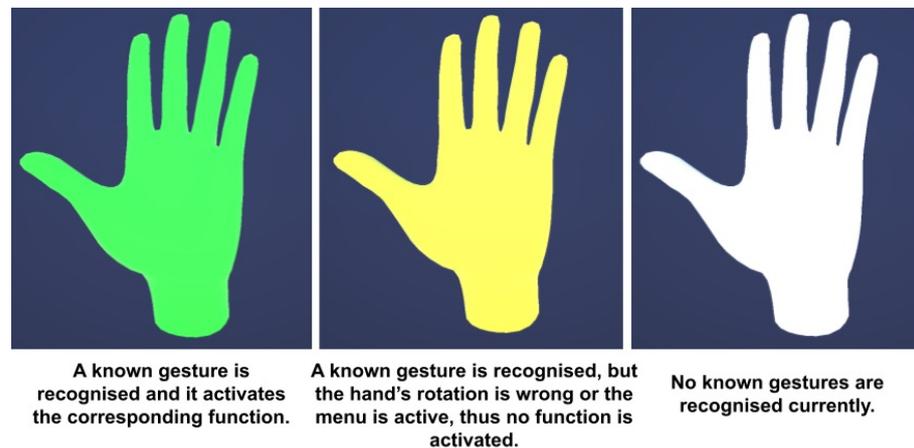


Figure 1. Gesture recognition color scheme to help test users understand when gestures are being recognised. Green colour indicates a recognised gesture which is currently activating a function. Yellow means a gesture is recognised but the hand rotation is wrong or the menu is active, resulting into no functionality. A white colour signifies that no gestures are recognised from the current hand pose.

Regarding the design of gestures, we identified three primary types of gesture controls based on their characteristics, as analyzed in a review by [42]: *Temporal*, *Contextual*, and *Instruction-based*.

Temporal refers to the gestures being *static* [43] (hand stays still) or *dynamic* [44] (hand moves from one gesture state to another). Static gestures are also known as *hand postures* or *poses*. As per our previous study [3], we again opted for mainly static gestures (although some of the gestures contain dynamic elements, such as aiming), as we believe that dynamic gestures suffer too much from the potential tracking issues and they will cause unnecessary physical strain for the user in addition to being difficult to perform for users with low hand dexterity levels [42,45].

Contextual gestures are categorized into two types: *communicative* and *manipulative*, as outlined in [46]. Communicative gestures are further subdivided into several classifications:

- *Semaphoric* [47,48]: These gestures form a unique language applicable specifically within the context of an application and unrelated to any real-world languages.
- *Pantomimic* [46,49]: These gestures replicate real-world actions or concepts.
- *Iconic* [50,51]: These refer to existing concepts, such as depicting the shape, location, or function of an object.
- *Metaphoric* [50]: These gestures represent abstract ideas or concepts.
- *Symbolic* [46,49]: Commonly recognized and understood gestures within a society, like the “thumbs up” sign signifying positivity in many Western cultures.
- *Modalizing symbolic/speech linked* [46,49,52,53]: Used in conjunction with speech, like simultaneously asking about someone’s spouse and gesturing widely to imply the spouse’s obesity.
- *Deictic* [47,49,51]: Gestures used for pointing or indicating direction.
- *Gesticulation* [51,52,54]: Minor gestures that enhance or support certain types of speech.
- *Adaptors* [55]: Unconscious gestures used to release body tensions like head shaking, and

- *Butterworth's* [56] gestures: They indicate that the person has lost their train of thought and is trying to recall a word (e.g., waving of a hand in place), although Butterworth's gestures have not been able to be replicated at the supposed frequency and might thus be misrepresented [44,57].

Our application utilizes mainly semaphoric gestures with deictic aspects in them, although it is difficult to avoid symbolism and iconism completely [42].

Manipulative gestures are employed to interact with and alter the (spatial) characteristics of an object, such as modifying its location or orientation in space [47]. Manipulative gestures play a significant role in this study particularly in tasks involving interaction with menus or 3D models from Task 8 onwards, as detailed in Sections 4.2 and 5.1. It is important to note that manipulative gestures can also have communicative functions in different contexts, and the reverse is true as well. For instance, in our system, the thumbs-up gesture in the two-handed user interface (2H UI) is semaphoric. Unlike its typical societal meaning of positivity, within our application, it forms part of a unique language. Here, the thumbs up, used alongside making a fist, is employed to regulate movement or manipulation speed, making it both semaphoric and manipulative in nature [42].

Instruction-based gestures consist of *prescribed* and *freeform* gestures. Prescribed gestures are part of a defined gesture language or library and cannot be altered by the user. They are designed to initiate a specific action without ambiguity or room for interpretation. On the other hand, freeform gestures are composed of a series of smaller movements combined to form a more complex action that is subsequently recognized as a specific function. Examples of these could include drawing a shape in the air, which then leads to the creation of an object resembling the drawn shape. Alternatively, an object might be moved in space by tracing a path with one's finger. These gestures are not preset and rely on the user's discretion and creativity. In our system, however, all gestures are strictly prescribed to ensure better comparability and consistency in their application and interpretation [42].

The reason for strict gesture choices, static (hand postures/poses) over dynamic, semaphoric with deictic aspects, and strictly prescribed, is also related to the scope of the study; it is just not feasible to test all gestures at once, so keeping the selection narrow allows the study to focus on the selection properly. After deciding on the initial gesture design, we conducted several pilot testing phases to ascertain that the gesture recognition level and intuitiveness were appropriate and that the tasks could be completed without major issues or software bugs. Based on the feedback received, the system was refined multiple times.

In conclusion, the selection of gestures for our study was guided by the previously mentioned key points and evaluated in light of the classifications within gesture theory. This process was informed by our expertise in user interface design for VR and insights gained from pilot testing. Furthermore, our review indicates that at present, there are no established collections of universally accepted gestures specifically for VR-based 3D design reviews or for gesture interaction in a broader context.

3.1. One-Handed User Interface

The one-handed user interface was developed with simplicity in the actions, action efficiency, and hand-switching/resting possibilities in mind. The gesture actions require only one hand to activate and stay active, leaving the other hand free to either rest or activate another action. An example for multi-interaction is to move forward and turn at the same time. Additionally, the movement actions, apart from turning, can be performed at double speed if both hands form the same gesture. While the UI is rather simple to use, its shortcomings include potentially more accidental action activation than in the two-handed UI. This happens because it is enough to have one hand in a correct position to trigger an action, and switching between actions without activating any others in the process can be difficult without practicing.

The possible user actions, their gesture types, and their acronyms in the 1H UI are explained in Table 1 while the gesture images can be seen in Figure 2.

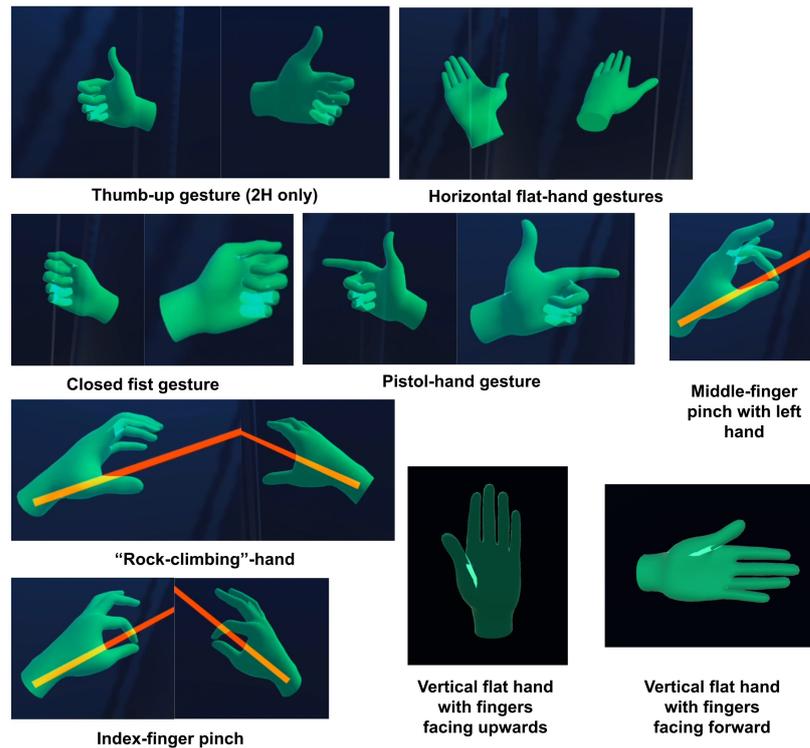


Figure 2. All gestures in the VR environment. The gestures are showcased in green, while the ray used for menu and teleportation is red. The 1H UI can use them all individually or combined except the thumb up, which is only for 2H. The 2H UI uses all gestures, but the non-pinch gestures are always accompanied by a fist or thumb-up gesture with the other hand.

Table 1. Gestures, their functionality, and gesture type in the one-handed user interface (1H UI).

Gesture	Functionality	Acronym	Type
Right/Left fingers forward facing flat vertical hand	Forward movement while recognized, direction controlled by headset gaze. Both hands doubles the speed.	FORW1	Unimanual (single hand), bimanual symmetric (both hands)
Right/Left fingers upwards facing flat vertical hand	Backward movement while recognized, direction controlled by headset gaze. Both hands doubles the speed.	BACK1	Unimanual (single hand), bimanual symmetric (both hands)
Right/Left horizontal flat hand with palm facing upwards	Vertical up-movement while recognized. Using both hands doubles the speed.	UP1	Unimanual (single hand), bimanual symmetric (both hands)
Right/Left horizontal flat hand with palm facing downwards	Vertical down-movement while recognized. Using both hands doubles the speed.	DOWN1	Unimanual (single hand), bimanual symmetric (both hands)
Right “pistol” gesture	Turns left while recognized, only one speed option.	TURNL1	Unimanual
Left “pistol” gesture	Turns right while recognized, only one speed option.	TURNR1	Unimanual
Right/Left “Rock-climbing” gesture	Brings up a visible forward-facing ray to the activating hand which follows hand movement.	RAY1	Unimanual

Table 1. Cont.

Gesture	Functionality	Acronym	Type
Left index-finger pinch	While aiming the left hand ray at a valid teleport location (ray turns green), teleports to it.	INDTELE1	Unimanual
Left middle-finger pinch	While aiming the left hand ray at a valid teleport location (ray turns green), teleports to it.	MIDTELE1	Unimanual
Left or right closed fist	If the right fist gesture is not active, the left fist activates the selection menu at the hand's location which follows the hand while the fist is recognized. Right fist disables the menu if the left fist gesture is not active.	MENU1	Unimanual (fist alone) and bimanual symmetric (one fist needs to be open and the other closed)
Right index-finger pinch	While aiming the right ray at the selection menu (ray turns blue), selects the currently hovered button.	SELECT1	Unimanual
Upwards pointing "pistol" gesture	Rotating a model on x or z axis toward the negative direction (or "counter-clockwise") depending on the activating hand, right hand rotates on x axis and left on z axis. Only one speed option.	UPTURNMOD1	Unimanual
Downwards pointing "pistol" gesture	Rotating a model on x or z axis toward the positive direction (or "clockwise") depending on the activating hand, right hand rotates on x axis and left on z axis. Only one speed option.	DOWNTURNMOD1	Unimanual

3.2. Two-Handed User Interface

The two-handed user interface was designed mainly with reliability and sense of control in mind. To achieve reliability, 2H gesture actions require coordination from both hands simultaneously for the movement actions, which rarely happens by accident. The actions are otherwise the same as in 1H, but the activating hand needs to be accompanied by the other hand in a fist or thumb-up position for an action to trigger or keep triggering. If the hand is in a fist, then the action is performed at normal speed (e.g., moving forward at normal speed: one hand is in fingers forward facing vertical flat hand and the other is in a fist), while in case of a thumb up, the action is executed at double speed (where applicable). There is also a mechanical addition to 2H which 1H does not offer: turning at double speed. The reason why 1H does not contain this feature is because pointing the "pistol" gesture to the same direction with both hands (which is how doubling speed works for other movement actions) is rather inconvenient and potentially even painful for extended periods. The menu and ray interaction (teleports, button/object selection) are otherwise the same except the menu is either enabled or disabled when both hands form a fist, which emphasizes the need for two hands in controls, even though the menu can still be interacted with just the right hand ray. This approach also potentially reduces accidental menu activation or disabling.

The possible user actions and their acronyms in the 2H UI are explained in Table 2, while the gesture images can be seen in Figure 2.

Table 2. Gestures, their functionality, and gesture type in the two-handed user interface (2H UI).

Gestures	Functionality	Acronym	Type
Right/Left fingers forward facing flat vertical hand combined with fist or thumb up on the other hand	Forward movement while recognized, direction controlled by headset gaze. Fist is normal speed, thumb up is double speed.	FORW2	Bimanual asymmetric
Right/Left fingers upwards facing flat vertical hand combined with fist or thumb up on the other hand	Backward movement while recognized, direction controlled by headset gaze. Fist is normal speed, thumb up is double speed.	BACK2	Bimanual asymmetric
Right/Left horizontal flat hand with palm facing upwards combined with fist or thumb up on the other hand	Vertical up-movement while recognized. Fist is normal speed, thumb up is double speed.	UP2	Bimanual asymmetric
Right/Left horizontal flat hand with palm facing downwards combined with fist or thumb up on the other hand	Vertical down-movement while recognized. Fist is normal speed, thumb up is double speed.	DOWN2	Bimanual asymmetric
Right “pistol” gesture combined with fist or thumb up on the other hand	Turns left while recognized. Fist is normal speed, thumb up is double speed.	TURNL2	Bimanual asymmetric
Left “pistol” gesture combined with fist or thumb up on the other hand	Turns right while recognized. Fist is normal speed, thumb up is double speed.	TURNR2	Bimanual asymmetric
Right/Left “rock-climbing” gesture	Brings up a visible forward-facing ray to the activating hand which follows hand movement.	RAY2	Bimanual asymmetric
Left index-finger pinch	While aiming the left hand ray at a valid teleport location (ray turns green), teleports user to it.	INDTELE2	Unimanual
Left middle-finger pinch	While aiming the left hand ray at a valid teleport location (ray turns green), teleports user to it.	MIDTELE2	Unimanual
Left and right closed fist simultaneously	If the menu is disabled, activates it and if the menu is active, disables it. The fists need to be reopened to trigger the functionality again.	MENU2	Bimanual symmetric
Right index-finger pinch	While aiming the right ray at the selection menu (ray turns blue), selects the currently hovered button.	SELECT2	Unimanual
Upwards-pointing “pistol” gesture combined with fist or thumb up on the other hand	Rotating a model on x or z axis toward the negative direction (or “counter-clockwise”) depending on the activating hand, right hand rotates on x axis and left on z axis. Fist is normal speed, thumb up is double speed.	UPTURNMOD2	Bimanual asymmetric
Downwards-pointing “pistol” gesture combined with fist or thumb up on the other hand	Rotating a model on x or z axis toward the positive direction (or “clockwise”) depending on the activating hand, right hand rotates on x axis and left on z axis. Fist is normal speed, thumb up is double speed.	DOWNTURNMOD2	Bimanual asymmetric

4. Materials and Methods

4.1. Methodology

In this study, a methodology similar to the Design Science Research Methodology for Information Systems Research (DSRM) by Peffers et al. [58] was utilized. The methodology is used for research which results in IT artefacts that are intended for solving identified organizational problems. This methodology consists of the following phases [58–60]: (1) Identify a problem, (2) Define objectives, (3) Design and develop, (4) Demonstrate the artefact, (5) Evaluate performance, and (6) Communication.

In this paper, the research problem and the objectives for our artefact are identified and defined in Section 1, while the pre-existing solutions are discussed in Section 2. Section 3 explains the design and development process of our artefact in addition to its complete properties. The demonstration of our artefact was completed with a case study of 25 test users in March–April 2022, the results of which are presented in Section 5 and discussed in Section 6. The test users were recruited using convenience sampling and they were mostly acquired among students and researchers at the University of Turku. This gives the test a bias toward these types of users. The demonstration utilized the Meta Quest 2 VR headset, and the hand-tracking was performed with the headset's four inside-out facing cameras, which mostly use visible light to approximate hand positions. Before the demonstration, each test user was given a written description of the test in detail, explaining what is required of them, what will be recorded and where it will be saved to (a secure cloud), and also that they are allowed to stop the test at any point if they so choose. This consent form was signed by each test participant so that informed consent could be ascertained.

For evaluating the demonstration performance, a self-designed testing questionnaire was used and filled in three parts: before testing, after testing the first UI, and after testing the second UI. The detailed testing phases and questionnaire categories can be seen in Tables 3 and 4. Standardized questionnaires such as the NASA Task Load Index (NASA TLX) [61], System Usability Scale (SUS) [62], and the Simulator Sickness Questionnaire (SSQ) [63] were considered but not deemed suitable for this study. The reasons for the NASA TLX were that there was no need to assess the specific load caused by the tasks but rather the controls were designed for them, whereas SUS, while potentially applicable, looks at system usability in a rather holistic manner, which is too broad a view for this study's aims. The reason to not include the SSQ was that it is not in the study's scope to thoroughly investigate what type of VR sickness occurred but rather to see if it occurred at all or not and at what frequency. Additionally, the whole testing experience was recorded using three high-resolution video cameras around the users with the help of spotlights for better visibility. The idea behind the recording was to see the hand gestures precisely in order to analyze them afterwards and compare them to the recordings made from within the VR applications. The recording also contained sounds in case the users needed to ask clarifying questions or would show any signs of fatigue, enjoyment, or discomfort. In addition to these measures, the applications logged the test users' gesture actions comprehensively. The logs were saved as text files and later parsed into Excel sheets via a script. The logging included the amount of activations and active durations of each gesture and action with the activating hand, the amount and duration when tracking of the users' hand(s) was lost for any reason, and task performance-related metrics such as completion time. The activation instances were additionally timestamped in order to be matched with the recordings later. With these metrics, it was possible to precisely analyze the usage for each gesture which could then be compared with the preference ratings from the questionnaire in order to understand more about gesture design for VR.

Table 3. Testing questionnaire categories, each category except background information is tailored for each version.

Category	Description
Background information	Previous VR and HT experience, age group, gender, handedness
Ease of use	Comparing the ease of use of different HT activities and potential hand preference to each activity
Ease of learning	Comparing the ease of learning of different HT activities and potential hand preference to each activity
Reliability of the functionality	Comparing the reliability of different HT activities and potential hand preference to each activity
Gestures	Questions about gesture preference when ignoring the actions they are connected to
Action speed and combining actions	Questions related to speed preference (normal or double) and potential use or need for combining different actions simultaneously
Physical and mental well-being	Questions related to VR sickness and the strain experienced in different body parts during testing

Table 4. Testing phases and durations.

Testing Phase	Description	Duration
Research introduction	Explaining the goals and data usage and asking written consent to the test	1–2 min
Filling background information part of the questionnaire	Demographics questions	1–2 min
Testing either one-handed or two-handed UI	Needs to complete all 13 tasks	20–30 min
The part of the questionnaire related to the just tested version	Answer all required questions	10–15 min
Testing the remaining UI	Complete the same 13 tasks again with altered controls	10–20 min
The remaining part of the questionnaire	Answer all required questions	5–10 min
Feedback section	Optional written and spoken feedback	0–1 min
Total	-	46–78 min

4.2. Testing Scenario

The testing scene, see Figures 3 and 4, was developed with the Unity game engine version 2020.3.6f1 [64] and the Oculus Integration software development kit v.38.0 [65]. The source code for the application is unavailable to the public for the time being. The scene consisted of a rectangular area which had opaque walls and floor, while the ceiling was left open, showing a default Unity skybox to create a sense of freedom for the users. The hand-tracking algorithm on the Unity side works by checking the difference between the user's hand position and any of the prerecorded gestures on the database on a finger bone-by-bone basis (bones are the animation rig bones created by the Oculus SDK). The differences between individual bones in the currently formed hand pose are calculated one at a time against the bones in the gestures saved in the database with a discard threshold of 0.09 (float). Exceeding the threshold for a finger bone in the database results in determining the hand that is not forming the corresponding gesture. If all the gestures in the list have been compared with and all have at least one bone exceeding the threshold, then the hand is determined to be not forming any gestures. If a match is found, meaning that the distance

of all current hand pose finger bones is less than the discard threshold from the gesture in the database, then the total combined difference of all the bones is summed up and saved as a new closest gesture. After all of the gestures in the database have been compared, the matched gesture with the lowest sum difference in the finger bone position to the current hand pose is determined to be the currently active gesture on the hand.

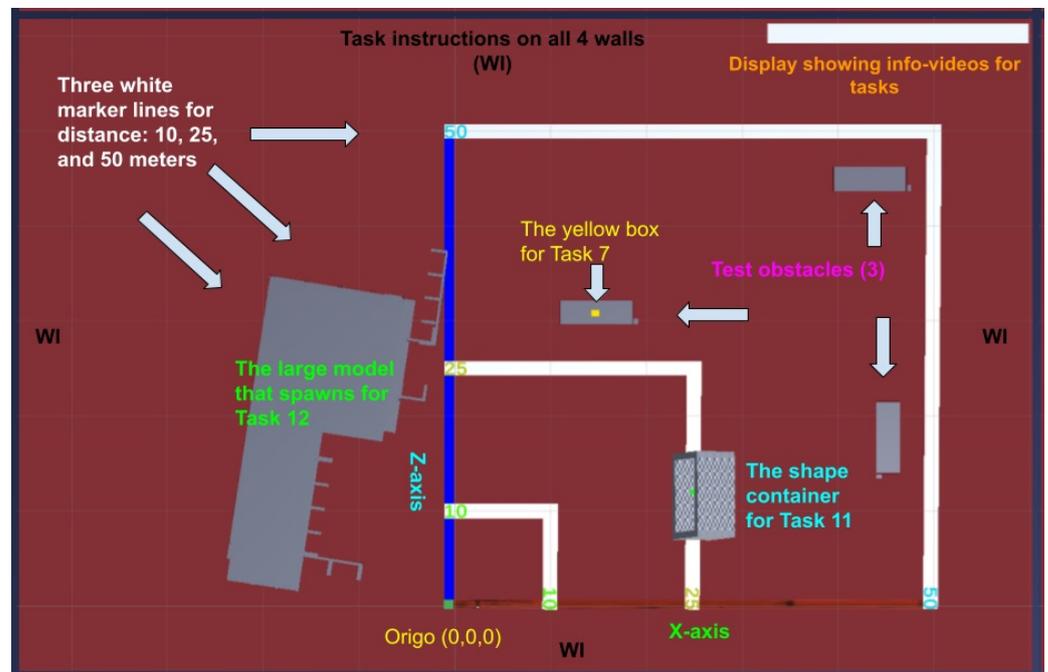


Figure 3. A top-down view of the testing scene, containing all relevant points of interest for the tasks used in the test.

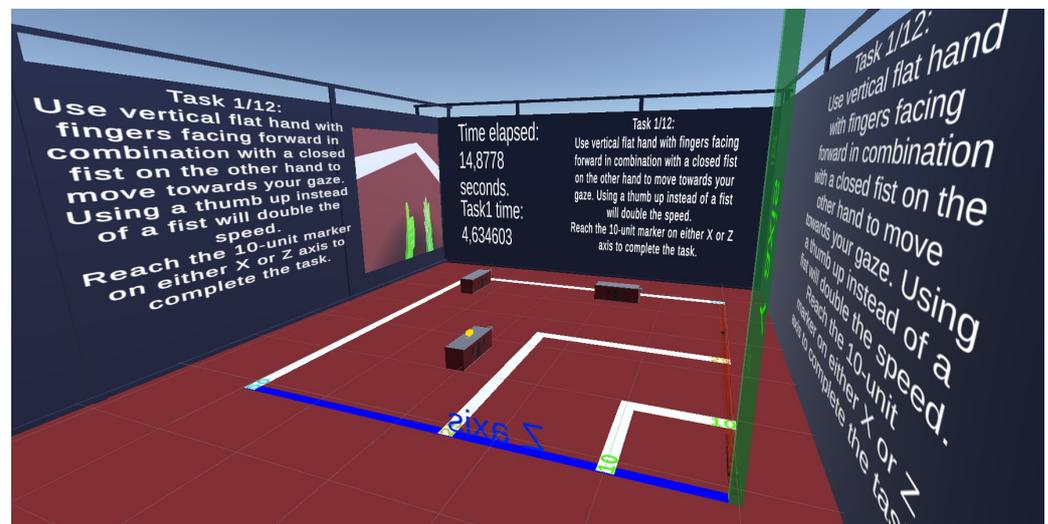


Figure 4. A sideways view of the testing scene, showcasing the timer and instruction systems for the users.

Both HT UIs were used to test 13 tasks in the testing environment, and counterbalancing was applied for the sample so that half tested the 1H UI first and the rest tested the 2H UI first. Each task contained written instructions in the test environment walls, as seen in Figures 3 and 4, as well as an instruction video showcasing the required actions. Additionally, on the walls were separate pictures of the gesture types which were the main focus in a given task. The actions unlocked one by one for the users; e.g., it was not possible to move

backwards (Task 2) until learning how to move forward first (Task 1). The instructions were displayed either in Finnish or English based on the test user's choice, although the instruction videos and gesture imagery contained no text so they were identical for all test users. The 13 tasks were divided into introduction (Task 0), learning how to move the user (Tasks 1–7), menu control (Tasks 8 and 9), model controlling (Tasks 10 and 11), and finally a “search and destroy” type of test to see how well the user remembers the movement controls and can apply them (Task 12). The task specifics are detailed in Appendix A. The main idea for the task design was to gradually increase the complexity of them by introducing mechanics one by one and then later on tasks where those mechanics needed to be applied, culminating into Task 12 where all mechanics were available.

Additionally, when designing the experiment, the difficulty and feasibility of the tasks were evaluated using heuristics similar to the ones by Sutcliffe and Kaur (2000), which propose the following usability-related questions for user tasks in virtual environments [66] (pp. 419–420): (1) Form goal, (2) Locate active environment and objects, (3) Approach and orient, (4) Specific action, (5) Manipulate object, (6) Recognize and assess feedback, and (7) Specify next action.

4.3. Result Calculation Rules

This section presents the test user categories and result rule calculation mechanisms. The test user categories and their acronyms based on the questionnaire answers can be seen in Table 5. The idea was also to evaluate differences between right-handed and left-handed users, but as there was only one left-handed user, the evaluation was not feasible.

Table 5. The test user categories.

Category Definition	Acronym	Users	Completed Testing
Little or no VR experience	NOVREXP	17	14
A lot of VR experience	VREXP	8	7
Little or no hand-tracking experience in both VR and AR	NOHTEXP	22	19
A lot of hand-tracking experience in either VR or AR or both	HTEXP	3	2
Tested one-handed version first	1FIRST	13	10
Tested two-handed version first	2FIRST	12	11
Male users (Gender)	MEN	13	12
Female users (Gender)	WOMEN	12	9
Between 18 and 35 years of age	YOUNG	21	19
Older than 35 years	OLD	4	2
Tested in Finnish language	FIN	8	7
Tested in English language	ENG	17	14
Aborted testing	ABORT	4	-
Completed testing	COMP	21	21
Total	ALL	25	21

The following rules were used to calculate preferences from the questionnaire:

For ease of use, ease of learning, and reliability, the users rated the gesture activities out of these options on a Likert scale:

- 1 = Very difficult to use/learn/Very unreliable;
- 2 = Quite difficult to use/learn/Quite unreliable;
- 3 = Hard to say;
- 4 = Quite easy to use/learn/Quite reliable;

- 5 = Very easy to use/learn/Very reliable.

These evaluations from the users were then averaged for the whole sample, which completed the testing (COMP), and divided further into the test user categories. The users who had to abort the testing (ABORT) were not included in any of the results except for the VR sickness evaluation. The test user categories were compared with their corresponding ones (e.g., users inexperienced with VR were compared with experienced VR users) in addition to comparing each category to the combined mean. The potential deviations from the expectations were noted and are discussed in their corresponding subsections.

In cases where there is a number in parentheses, e.g., (2.5), without additional explanation, it refers to a mean value of the currently discussed value type and category, e.g., ease of use for the test users with a lot of prior VR experience (VREXP). As per the aforementioned ratings, the mean values range from 1 to 5.

The users' hand preferences for actions were also evaluated with options right, left, or no preference. The idea was to see if any actions were easier to use/learn or more reliable with either hand. The evaluated activities were the ones which afford using either hand, meaning only the user and model movement actions, not including menu controls which are tied to a specific hand. For the hand preference questions, the turning movements to either direction were combined into one activity, so TURNL/TURNR and UPTURNMOD/DOWNTURNMOD, to simplify the choosing process and also because there was no option to turn left with the left hand and vice versa. Additionally, ray action preference by hand was queried.

The users' gesture activations, the amount of activations and active durations for a specific hand per all possible gestures and actions for it were logged by the application. They were calculated using the following rules.

For both hands individually, the amount the hand activated a gesture and the duration it kept a gesture active were calculated. There were six gestures in total: "flat hand" (with no specification for the hand rotation), "pistol" (pointing to any direction), closed fist, thumb up (for 2H only), index-finger pinch, and middle-finger pinch. Additionally, if the gestures activated any actions, the amount of activations and active durations for those were logged as well (see Section 3 for action definitions). The gesture/action activation amounts are primarily utilized to see if there were a lot of difficulties in activating an action or if some gestures/actions were accidentally activated often, whereas the active duration shows the utilization preference with a higher reliability level. However, if only the exact numbers from the logs would have been used to compare action or hand preferences, the results would be skewed by the most and least efficient users. This is why an additional relative value was calculated. The relative number compares all the potential actions for the user during a given time/task and then calculates a percentage value for how popular a certain action was for the user at the time. Using this calculation mechanism, one can easily see what actions or gestures were preferred in any given task with just one variable.

The equation for the calculation of the relative activation amount (or the active duration) is as follows, where:

- $T_n\%$ = the relative activation amount for the action n ;
- T_n = the absolute activation amount for the action n ;
- C_{all} = the combined absolute activation amount of all possible actions currently available, the amount of which changes based on the current task.

$$T_n\% = \frac{T_n}{C_{all}} \quad (1)$$

This relative calculation method was used for gestures and actions, while a similar method for the task duration estimation was used. The task duration method takes the absolute time value of a task, e.g., 1 min 17 s, and divides that with the total time the version took to test, e.g., 30 min, for every user individually. Using this method and averaging the relative numbers of each user, one will arrive at a percentage value for each task which

showcases how long it took (or how difficult it was) to complete in comparison to the users' average task clearing speed. Additionally, for some actions, there were two speed options: normal and double speed, which were calculated jointly, e.g., 60% of the forward movement happened with normal speed and the remaining 40% was double speed in a task where only forward movement was allowed. In a task with also backwards movement, the statistics could be similar to the following: 40% FORW normal speed, 20% BACK normal speed, 25% FORW double speed, and 15% BACK double speed, which equal 100%.

4.4. Statistical Tests

T-tests (or Student's *t*-tests) [67] were utilized for testing statistical significance in our data. When comparing test user groups, such as males and females, independent two-sample *t*-tests were utilized in the following way. First, a normal distribution of the data was tested with a Shapiro–Wilk test [68] using a significance level of 0.05, and then an assumption about equal variances was made using a Levene's test [69] with the same significance level. Finally, the *t*-test was calculated using an alpha of 0.05 (95% significance level).

The same method was applied for variable comparisons in one version, such as right-handed gesture usage versus left-handed gesture usage in the 1H UI, with the addition of calculating the Pearson correlation coefficient [70] where applicable and deemed necessary. For comparing variables between versions, e.g., the 1H UI completion time versus the 2H UI completion time, a paired *t*-test was conducted instead.

In cases where the data were not normally distributed, the means, standard deviations, and any potential qualitative data were utilized instead. The Accord.NET library [71] was utilized for the statistical test functions.

4.5. Hypotheses

The main hypotheses that were explored are outlined below.

Hypothesis 1. *The 1H UI will be completed faster than the 2H UI on average.*

The first hypothesis is based on the possibility of combining actions in the 1H UI (e.g., being able to move and turn simultaneously). In particular, Tasks 11 and 12, which require a lot of different actions, could be faster on average due to this.

Hypothesis 2. *The 2H UI will be rated as more reliable to use and will have less accidental gesture activation reports.*

The second hypothesis is based on the notion that it is easier to control when actions are initiated with the 2H UI.

Hypothesis 3. *The test users with a lot of VR experience will on average complete the test faster than less experienced users, and additionally, they will report less VR sickness. Similarly, the test users with a lot of hand-tracking experience will be faster than less experienced test users even if their VR experience is on par.*

The third hypothesis relates to the assumption that more experience in a subject equals better performance.

5. Results

5.1. Results Based on Task Performance

5.1.1. Task Completion Times

The detailed task descriptions and individual task-based results can be found in Appendix A. The completion time for each task was logged for each user to see which tasks were faster or slower than others and how much the version or test user traits impacted on the time. The results, shown in Table 6, show that the 1H UI was somewhat faster

than the 2H UI overall, and most of the difference was made in the last three tasks, 10–12, whereas the 2H UI was faster in Tasks 0, 3, 4, 7, 8, and 9. Moreover, the starting Task 1 was completed in around half the time in the 1H UI compared to the 2H UI, indicating that starting the testing was a lot more difficult in the 2H UI, and thus, it was more difficult to learn. The Task 1 result is partially to be expected, as the control scheme for the 1H UI was designed to be easier to initiate, although the completion speed should have ideally picked up by the time the later tasks were started as the users became accustomed to the controls.

Table 6. Completion times per task for the users who completed testing. The s-% means the relative value of standard deviation (s) divided by the mean for a given task. % refers to a task's relative completion time compared to the total test completion time on a tester by tester basis. The highest values are highlighted in red, while the lowest values are highlighted in blue.

One-Handed UI							
Task	μ	Mdn	Min	Max	s	s-%	%
Task 0	2 min 14.4 s	1 min 29.4 s	29.0 s	8 min 59.4 s	2 min 2.2 s	91%	9%
Task 1	23.4 s	16.2 s	7.1 s	1 min 16.3 s	18.3 s	78%	2%
Task 2	1 min 5.8 s	34.2 s	2.4 s	5 min 38.6 s	1 min 27.3 s	133%	5%
Task 3	28.6 s	17.4 s	6.5 s	2 min 18.8 s	30.7 s	107%	2%
Task 4	13.3 s	8.1 s	3.6 s	1 min 2.6 s	14.5 s	109%	1%
Task 5	29.8 s	23.8 s	15.3 s	1 min 40.0 s	18.1 s	61%	2%
Task 6	1 min 0.3 s	57.3 s	12.9 s	1 min 52.8 s	31.9 s	53%	5%
Task 7	1 min 16.0 s	1 min 2.4 s	14.1 s	3 min 1.2 s	53.8 s	71%	6%
Task 8	1 min 51.6 s	1 min 9.0 s	33.2 s	8 min 34.8 s	1 min 55.2 s	103%	7%
Task 9	2 min 29.6 s	1 min 12.5 s	20.1 s	8 min 6.5 s	2 min 32.0 s	102%	9%
Task 10	4 min 51.7 s	3 min 31.4 s	1 min 3.6 s	16 min 37.5 s	3 min 40.7 s	87%	20%
Task 11	5 min 34.8 s	3 min 23.5 s	30.0 s	20 min 28.2 s	5 min 37.9 s	101%	19%
Task 12	3 min 24.3 s	2 min 33.7 s	55.6 s	9 min 12.5 s	2 min 16.9 s	67%	14%
Total	25 min 23.7 s	20 min 28.7 s	6 min 51.3 s	1 h 1 min 52.0 s	14 min 49.1 s	58%	100%
Two-Handed UI							
Task	μ	Mdn	Min	Max	s	s-%	%
Task 0	2 min 9.0 s	1 min 36.6 s	33.4 s	4 min 53.0 s	1 min 13.4 s	57%	8%
Task 1	58.1 s	40.1 s	12.7 s	2 min 6.5 s	36.7 s	63%	3%
Task 2	1 min 26.0 s	36.8 s	10.3 s	6 min 55.1 s	1 min 59.7 s	139%	4%
Task 3	28.5 s	20.8 s	5.4 s	1 min 12.4 s	20.0 s	70%	2%
Task 4	12.8 s	10.7 s	5.9 s	30.2 s	6.7 s	52%	1%
Task 5	28.1 s	25.5 s	12.4 s	58.2 s	12.4 s	44%	2%
Task 6	1 min 35.9 s	1 min 0.3 s	14.4 s	5 min 8.5 s	1 min 24.0 s	88%	5%
Task 7	1 min 7.7 s	1 min 3.2 s	11.9 s	2 min 59.7 s	46.8 s	69%	4%
Task 8	1 min 9.7 s	1 min 8.2 s	17.3 s	2 min 38.1 s	35.7 s	51%	4%
Task 9	2 min 5.2 s	1 min 22.0 s	24.9 s	9 min 52.9 s	2 min 11.4 s	105%	7%
Task 10	7 min 40.3 s	4 min 7.3 s	1 min 21.5 s	20 min 10.2 s	6 min 23.9 s	83%	22%
Task 11	7 min 36.6 s	3 min 41.8 s	36.2 s	24 min 49.2 s	7 min 18.6 s	96%	20%
Task 12	5 min 35.0 s	3 min 56.6 s	56.7 s	18 min 23.0 s	4 min 30.4 s	81%	18%
Total	32 min 33.1 s	29 min 51.0 s	8 min 30.6 s	1 h 5 min 9.3 s	19 min 53.4 s	61%	100%

Tasks 10 and 11 took the most time on average in both versions which was expected, as the later tasks were more complicated than the early ones, even though the users had more experience by the time they reached the later ones. However, it is clear that Task 12 was somewhat easier than 10 and 11 for both versions, despite similar difficulty design, indicating that model moving had challenges. The quickest task was clearly Task 4 for both versions; thus, it was likely the easiest, whereas Task 0 and Task 2 took considerably more time than optimal. Task 0 was understandably slow, as it was the first touch to the virtual environment, and there were a lot of instructions to read which took a varying amount of time to understand for different test users. Additionally, some users had to spend a lot of time fitting their glasses into the headset. Conversely, Task 2 should not have been difficult, but as noted in Appendix A.2, there were major learning difficulties for many users. Furthermore, the relative standard deviation was clearly highest in Task 2 in both versions (>130%), showcasing further that some users understood the task and completed it quickly, whereas some did not understand it at all and potentially were lucky to make it through eventually.

In the 1H UI, the fastest test user (6 min 51 s) had already completed the whole test when another user was still reading instructions in Task 0 (8 min 59 s), which showcases the extreme variance between users. Although this is an extreme example due to a single very slow user and a very fast user, the overall maximum and minimum time comparisons for tasks show a great difference in both versions. However, this result needs to be inspected through the lens of testing order, as there was a major difference in the completion speed based on whether a version was tested first or second. The result for the 1H UI was 29 min and 8 s when tested as first, and it was 20 min and 46 s when tested as second, whereas for the 2H UI, it was 46 min and 6 s when tested as first and only 17 min and 39 s when tested as second. The result was spread out considerably more for the 2H UI, indicating that it was a lot more difficult to learn as a first experience, but as the second, it was faster than the 1H UI, suggesting that with sufficient experience, it could become more efficient to use.

Overall, Hypothesis 1 (see Section 4.5) can be considered proven, as the 1H UI was clearly faster on average, and specifically so in Tasks 10, 11 and 12, although this was not related to the action combining possibility, which was hardly utilized. Additionally, the VREXP users were considerably faster (1H = 21 min 34 s, 2H = 18 min 18 s) than NOVREXP users (1H = 24 min 56 s, 2H = 37 min 52 s), and the HTEXP users were faster (1H = 24 min 1 s, 2H = 13 min 30 s) than the NOHTEXP users (1H = 26 min 11 s, 2H = 34 min 33 s). Thus, the second part of Hypothesis 3 can be supported, resulting in it being supported completely.

The statistical analysis with t-tests described in Section 4.4 based on VR experience, HT experience, and testing order in relation to completion time showed that there was a statistically significant difference between the completion time and testing order for the 1H UI ($p = 0.016$) as well as the 2H UI ($p = 0.0002$). For VR experience, there was no statistical significance in the 1H UI ($p = 0.72$), while the 2H UI data were not normally distributed enough (Shapiro–Wilk $p < 0.05$). For the HT experience, there were not enough HT experienced test users to assess data normality (<4); thus, the t-test could not be conducted. Thus, it can be said that the difference in testing order resulted in larger and statistically more significant differences than previous VR or HT experience, which was partly also due to the low number of test users with a high amount of VR or HT experience. Additionally, the tester age seemed to be a factor as well, but there were not enough OLD users to test for statistical significance. To conclude, we can rule that the testing order was the greatest influence over completion time based on the measured variables and that it was the right decision to counterbalance it by dividing the test users in half for it.

5.1.2. Combined Task Results

This section assesses all tasks (detailed in Appendix A) jointly for the measurable performance aspects. The relative movement actions statistics, see Table 7, reveal that the forward movement was overall the most utilized action, and it was mostly used with

normal speed. The result was to be expected, as the forward movement is a familiar action from the real world and it was the first action the users learned. The backwards movement, however, was utilized only a little despite it being taught early on with just less than 10% overall utilization in both versions. This could be due to moving backwards being rare in the real world. For the vertical movement options, downwards movement was slightly preferred in both versions, but the extent was considerably more noticeable in the 1H UI. A large reason for this was related to the accidental teleportation, which caused many test users to end up high up in the air, thus requiring a descent. Additionally, there was no discernible usage of action combining in the 1H UI, and normal speed usage was favored overall.

Table 7. Relative action usage over the whole test duration (task independent), calculated with the relative calculation mechanism described in Section 4.3. The highest value(s) for each version in both action speeds is highlighted in red, while the lowest value(s) is highlighted in blue.

Action	Relative Activations				Relative Active Time			
	Normal Speed		Double Speed		Normal Speed		Double Speed	
	1H	2H	1H	2H	1H	2H	1H	2H
FORW	40%	35%	11%	21%	32%	40%	12%	14%
BACK	7%	5%	2%	3%	6%	6%	2%	2%
UP	7%	6%	2%	3%	7%	8%	3%	3%
DOWN	16%	8%	4%	5%	12%	9%	7%	3%
TURNRL	10%	8%	-	5%	18%	12%	-	4%

Out of the gestures, see Table 8, the “rock-climbing” gesture emerged as the most popular overall, while “pistol” was utilized the least. The high rate of utilization of the closed-fist gesture in the 1H UI indicates some issues with the menu and/or gesture recognition in general, as the fist was strictly necessary only in Tasks 0, 8, 9, and 10 of which 0 and 8 were quite short and 10 barely required it. For pinch gestures, the index-finger pinch was utilized in around 75% of the activations, which clearly confirms that it was the preferred gesture, and even though the menu interaction required the index-finger pinch, the teleportation results closely reflect this sentiment with double the amount of INDTELEs than MIDTELEs in both versions. Furthermore, MIDTELE peaked at its introduction in Task 6.

Table 8. Relative gesture usage over the whole test duration (task independent), calculated with the relative calculation mechanism described in Section 4.3. The highest value(s) for each version in both gesture categories is highlighted in red, while the lowest value(s) is highlighted in blue.

Non-Pinch Gestures	Relative Activations		Relative Active Time	
	1H	2H	1H	2H
Closed-fist gesture	21%	25%	25%	34%
“Flathand” gesture	24%	15%	28%	22%
“Rock-climbing” gesture	43%	31%	36%	25%
“Pistol” gesture	11%	6%	12%	9%
Thumb-up gesture	-	23%	-	9%
Pinch gestures	1H	2H		
Index-finger pinch	76%	78%		
Middle-finger pinch	24%	22%		

The overall hand preference ratios, according to both the questionnaire and the performance, indicate that the 1H UI was strongly preferred with the right hand, while the 2H UI

had a more even preference between hand usage, although many gestures were preferred with the left hand. It is difficult to determine for sure, as all but one of the test users were right-handed, but the result indicates that single gesture controls are potentially preferred with the dominant hand whereas joint controls can be more varied. Additionally, there was no proof that the dominant hand would activate more gestures than the recessive one.

In relation to menu usage, there were no major differences between versions for button selection. For ray activation, the right hand was preferred in the 1H UI, while the 2H UI was more varied, as prior results indicated. The menu was activated 53.9 times for 6 min and 29 s in the 1H UI and 49.6 times for 7 min and 33 s in the 2H UI. This could mean that the 1H UI had more issues with accidental menu activation, as there were more activations in relation to the active time, which indicates more quick open and close actions.

When it comes to the model controlling, the y axis turning was favored out of the possible actions in both versions with no major hand preference. When turning the model on the x or z axis, the UPTURNMOD action was utilized around four to five times more than DOWNTURNMOD, which indicates that pointing the “pistol” gesture upwards is more convenient for the users than pointing it downwards. Similar to user movement, the model was mostly moved forward at normal speed, and there was little vertical or backwards movement.

There was a lot of variance between different user categories, although no clear correlations apart from the testing order results, which show that testing a version as second was a lot easier to use and learn, more reliable and also quicker. In general, the OLD users had more issues than the YOUNG users, but their sample size was very small (two completed users). Additionally, the 2H UI was preferred by most users only in the category which tested it as second, further confirming that the testing order had a huge impact. The reason is likely due to learning the required tasks and the general adjusting period of VR and HT when testing the first version; thus, during the second version, the previous experience impacts more than the difference between the control schemes. The testing order was especially visible with the completion times in the 2H UI (see Section 5.1.1), which indicates that the 2H UI could become more usable with more experience using the system.

5.2. Questionnaire Results

5.2.1. Ease of Use

This category contained questions related to ease of use of the gesture actions, which were previously specified in Tables 1 and 2. These results only contain COMP users. Overall, the 1H UI was virtually as easy to use (3.9) as the 2H UI (3.8). The results also showcase that both UIs were quite easy to use on average, meaning that the design was probably somewhat successful. When considering the individual gesture actions, the easiest action turned out to be FORW1 (4.5) and UP2/DOWN2 (4.4), whereas the most difficult actions were MENU1 (3.1) and DOWNTURNMOD2 (3.4). Moreover, moving forward was rated considerably easier (1H = 4.5, 2H = 4.1) in both versions than moving backwards (1H = 4.1, 2H = 3.7), which indicates that backwards movement was perhaps difficult to use despite being functionally similar to forward movement, which was considered among the easiest actions. This can be due to many users failing to adequately learn the function in Task 2.

When looking at the different test user categories, a major influence on the results was the testing order: the version tested first was rated significantly more difficult (1H = 3.6, 2H = 3.5) to use than the version tested second (1H = 4.1, 2H = 4.1). This rather strongly indicates that the testing order made a large difference to the perceived ease of use, which was maybe even larger than the mechanical differences between versions. Additionally, OLD users rated the 1H version as considerably more difficult (3.3) than the YOUNG users (3.9), indicating that older age correlates with higher control need for actions.

5.2.2. Ease of Learning

The ease of learning (for COMP users) in general was rated 4.0 in the 1H UI and 3.7 in the 2H UI, showing a slight edge for the 1H UI. The easiest to learn actions were FORW1 (4.6) and DOWN2 (4.2), while the most difficult actions were UPTURNMOD1 and DOWNTURNMOD1 (3.2) in addition to DOWNTURNMOD2 (2.9). As the highest and lowest ratings were higher in the 1H UI than in the 2H UI; this indicates higher learnability in the 1H UI.

In the 1H version, the NOVREXP users learned more easily than VREXP users, while in the 2H UI, the result was the opposite, which indicates that the 1H UI was more intuitive as even inexperienced users could learn it quickly. The testing order had a similar impact on the ease of learning than on the ease of use, meaning the versions tested first performed a lot better. The test age had an impact as well, as the YOUNG users rated learning much easier than the OLD users in both versions.

5.2.3. Reliability

Overall, the reliability was rated equally at 3.8 in both versions, which disproves the first part of Hypothesis 2. Additionally, the users' own movement actions were rated as more reliable than menu and ray usage, which was observed as a lot of involuntary menu activation. The most reliable activities were FORW1 (4.5) and DOWN2 (4.3), while the least reliable were MENU1 (2.7) and MENU2 (3.0), further highlighting the difference in reliability between movement actions and menu controlling. The teleportation types showed no differences in reliability ratings, which is in contrary to observations which indicate that MIDTELE was less reliable. Moreover, higher levels of prior VR experience correlated with higher ratings of perceived reliability in most activities for both versions, and older tester age correlated with lower ratings.

Similarly to ease of use and learning, the testing order impacted reliability as well, meaning the version tested first was perceived as a lot more reliable than that tested second. Thus, the conclusion that the testing order greatly determined the perceived experience can be validated.

5.2.4. Gesture Preference

The test users were asked to name three of the most and least pleasant gestures (disconnected from their functionality) in no specific order. The results, showcased in Table 9, show that the "flathand" with fingers facing forward gesture was clearly picked as the most pleasant gesture on its own and also when joined with a closed fist or a thumb up. The clearly least pleasant single gesture was the "rock-climbing", which was likely due to its observed accidental activation. The closed fist and both fists closed gestures had the most variance among the gesture types, which indicates that perhaps the fist was convenient for some of the users, which was maybe due to reasons such as hand size or default posture, while a similarly large number of users had major difficulties with it.

Furthermore, the "pistol" gestures were rated as rather unpleasant as both single and joint gestures, although pointing up or down was considerably more unpleasant than pointing left or right. Additionally, the middle-finger pinch was rated less preferred than the index-finger pinch overall, confirming the observations about users having difficulties activating it.

Testers rated the tendency of accidental gesture recognition with a 1-unit interval Likert scale from 0 (gestures never activated accidentally) to 10 (gestures constantly activated accidentally) as 6.3 (1H) and 5.9 (2H), which indicates that the more reliability-focused 2H UI was able to give out a small perception of increased control. Thus, although there were no clear reliability differences as noted in Section 5.2.3, the accidental recognition result suggests that Hypothesis 2 might be somewhat accurate at least. Among the test user categories, there was a large disparity between the experienced HT users who rated lower (1H = 5.0, 2H = 4.0) compared to the non-experienced (1H = 6.4, 2H = 6.1). Additionally, the OLD users experienced a lot more accidental gesture activation (1H = 9.0, 2H = 7.0) than

the YOUNG users (1H = 6.0, 2H = 5.7). Positioning one's hands in parallel to one's legs near the hips was a common recommendation by the users to prevent gestures from being registered. When asked to rate the hand preference for each gesture, the right hand was overall preferred in both versions over the left hand, although this was a lot more so in the 1H UI, and there was considerably more indifference about the hand preference in the 1H UI than in the 2H UI. This indicates that the gestures in the 2H UI should perhaps be designed to be used with a specific hand combination.

Table 9. The relative values of the three most and least pleasant single (in 1H UI) and joint (in 2H UI) gestures when separating the gestures from their functionality. The 100% result means that everyone chose this gesture as one of the three most or least pleasant gestures, while 0% means that no one did so. The highest value(s) in each category is highlighted in red and the lowest value(s) is highlighted in blue. Abbreviations: MP = most pleasant, LP = least pleasant.

Gesture	MP Single	LP Single	MP Joint	LP Joint
"Flathand", fingers forward	71%	0%	76%	14%
"Flathand", fingers upwards	19%	0%	33%	14%
Closed fist (single) / Both fists (joint)	33%	38%	38%	48%
"Flathand", palm upwards	57%	5%	57%	10%
"Flathand", palm downwards	57%	14%	52%	5%
"Rock-climbing"	0%	43%	-	-
"Pistol", pointing left	14%	33%	19%	38%
"Pistol", pointing right	14%	38%	10%	43%
"Pistol", pointing up	5%	38%	10%	62%
"Pistol", pointing down	5%	38%	5%	67%
Index-finger pinch	19%	19%	-	-
Middle-finger pinch	5%	33%	-	-

5.2.5. Action Speed and Combining Actions

The movement actions were able to be executed with either normal or double speed. The normal speed was greatly preferred in model controlling actions for both versions, while double speed was preferred for the user movement actions except for the turning. This result reflects that controlling the model was more difficult than controlling oneself, as the model was moved at lower speed to lessen the impact of mistakes. Additionally, the double speed turning option was not deemed necessary.

Many users did not realize that combining actions was possible (29%), and the ones who did, for the most part (57%), barely utilized the feature. Only a few test users (14%) combined actions sometimes, and none claimed to have combined them often. Moreover, the possibility was not deemed necessary in the 2H UI either when polled. These results indicate that the possibility to combine actions is such an advanced feature that it will need a proper adjustment period to the system before its usability can be determined. Additionally, they mirror the task-based results showcased in Section 5.1.

5.2.6. Strain and VR Sickness

This section reviews the experienced strain and artificial movement-induced VR sickness statistics from the questionnaire among all test users, also including the ones who had to abort testing. These questions were posed because there is a need to understand how much strain gesture user interfaces like this might cause in the long run and whether they could alleviate VR sickness due to hand gestures being better connected to the movement functionality and being able to provide physical cues for the body. This potentially lessens disorientation occurring due to visually induced experiences of self-motion, *vection* [40],

which is caused by artificial movement in VR. This is further tested with changing direction with the normal and double speed options, as this can further increase the sickness occurrences [41]. Avoiding VR sickness is most important in any setting but especially when adopting VR technology for industrial use, as potential changes to existing processes will be costly and require a long adjusting period. Thus, it is important that the first experience is not a very negative one at least, which requires not experiencing strong VR sickness. In summary, while strain during use is important, it is not something that is immediately evident, unlike VR sickness usually is, so for that reason, sickness mitigation should take precedence in the UI design.

The participants were asked to rate their experienced VR sickness levels for each UI and separately for each movement mechanic within it. The results showed rather mild levels of VR sickness overall (1H UI = 2.0 and 2H UI = 1.7), as seen in Figure 5, but four test users needed to abort the testing due to feeling too nauseous. Three of the users aborted in the 1H UI (and had not tested the 2H UI prior), while one user aborted in the 2H UI (without testing the 1H UI first). Three out of the four users decided to stop on their own, while one of them was forced to stop due to vomiting suddenly. The highest level of sickness rated by those who experienced any was experienced during the teleporting and turning in both UIs, although the 2H UI was rated to have caused slightly stronger sickness by the ones who experienced any, whereas overall, it was less sickness inducing. Additionally, as per Hypothesis 3, the NOVREXP users rated considerably higher VR sickness than the VREXP users, proving the first part of the hypothesis. The same did not apply for HTEXP and NOHTEXPT users though, as the result was inconsistent between versions. Furthermore, the users who experienced any amount of sickness were asked about the impact of movement speed on the VR sickness with the majority (1H = 57%, 2H = 78%) saying that faster movement increased the sickness levels, which has some backing in the literature [72]. Interestingly, there were some users who felt that faster speed actually lowered their sickness levels (1H = 18%, 2H = 22%).

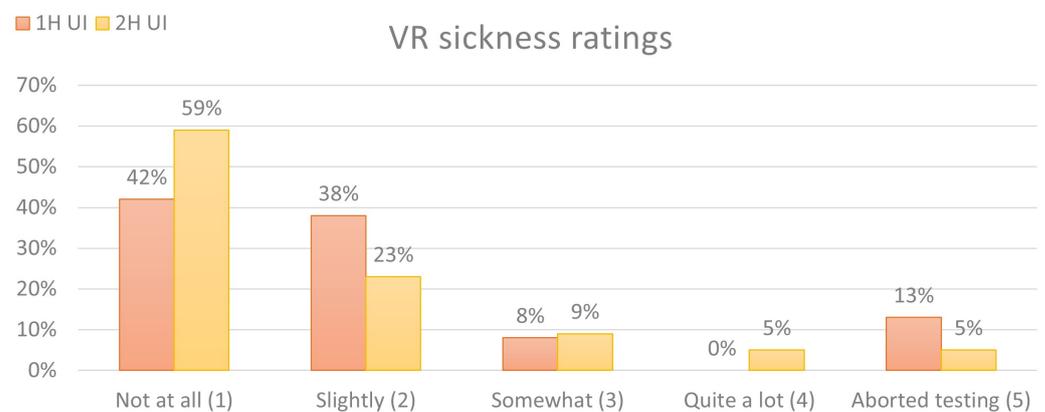


Figure 5. The VR sickness ratings from the questionnaire for all 25 test users (also the ones who aborted unlike in other statistics).

The strain experienced in different body parts (hands, fingers, wrists, shoulders, arms, and neck) was rated low (1.5/5 in both versions, which is between no strain (1) and light strain (2)) and not focused on a specific body part. When asked about strain in any other body parts, many wrote that their forehead had hurt because of the headset or that their face became uncomfortably sweaty. Clearly, the headset comfort issues are still present when utilizing VR, which might be a further deterrent for frequent usage. All in all, the VR sickness seemed to occur on a user-by-user basis, and it was difficult to determine whether either of the control schemes had an impact on it. Likewise, the experienced strain was low and there were no clear signs or feedback to suggest that either version was more or less strenuous than the other.

5.2.7. Version Preference

The results, showcased in Table 10, show that clearly the 1H UI was preferred in almost all user categories, while the 2H UI was only preferred when tested as second, showcasing the impact of the testing order. Overall, the preference ratio was 67% 1H UI and 33% 2H UI, which at least based on user preference confirms that the 1H UI, and thus the unimanual gesture type, was more enjoyable to utilize.

Table 10. Preferred version per tester category, highest occurrence is highlighted in red per category. Also shows the distribution of those who could not rate due to having to abort the testing.

Tester Category	One-Handed Version	Two-Handed Version	Testing Aborted (N/A)
NOVREXP	9	5	3
VREXP	5	2	1
NOHTEXP	12	7	3
HTEXP	2	-	1
1FIRST	4	6	3
2FIRST	10	1	1
MEN	9	3	1
WOMEN	5	3	3
YOUNG	13	6	2
OLD	1	1	2
FIN	6	1	1
EN	8	6	3
ALL	14	7	4

6. Discussion

The study indicates that gesture-based user interfaces in VR will likely require an adjusting period from the users before they become usable; thus, it is difficult to determine reliable usability levels by only utilizing inexperienced users. We suggest that a test should start with a neutral environment where the users become acquainted with being in VR and seeing their hands form gestures before giving the gestures functionality. This approach would further reduce the impact of the testing order when comparing multiple similar versions of a mechanic, which our test showed having a significant impact on learnability and thus preference. Furthermore, we found an indication about potential better usability or at least preference of unimanual gestures unlike, e.g., research by [25], which was inconclusive on the matter.

In terms of VR sickness, our hypothesis was that more realistic functionality, in our case utilizing one's hands to form gestures which better represent the VR movement functionality, would reduce the sensory conflicts between experienced (physical) and visually induced motion. As there were four users who experienced strong levels of VR sickness, one even vomiting, it can be said that gesture–interface use alone unlikely removes VR sickness completely. However, further testing is required to determine whether it can be reduced using this method. Our mechanic can also fall prey to the uncanny valley effect of virtual interactions, medium-level interaction fidelity [39], as while the hand-tracking system mirrors the users' hand movements rather precisely apart from slight latency, tracking occlusion and range issues, it still does not match real life motion perfectly. To become closer to high-fidelity interaction and out of the valley, the gestures could be designed to be dynamic, further reinforcing the users' senses about the correlation between visual and experienced motion. However, this approach is difficult to implement, as the tracking range of the cameras is limited, and moving the hands a lot during gesture formulation will incur tracking errors. Furthermore, constant hand movement is taxing

physically for the user, which will hamper the user experience in other ways even if it would succeed in reducing VR sickness. Additionally, most users felt that slower movement speed reduces VR sickness (as prior researched [72]), although as the result was not unanimous, it would be interesting to test the speed's impact with more options than just two.

In relation to the mechanics we used, the reason for using ray-based interaction with the menu instead of physics-based was partially due to a previous experiment's requirements [3] on which the system was built in addition to technical issues with hand-tracking creating colliders in the Oculus SDK [65], which initially made it less functional. For further studies, testing with a physics-based menu and buttons might make the system more intuitive, as physical button-pressing, even without haptic feedback, has considerably higher biomechanical symmetry [39] than the ray-based pinching.

Additionally, the tasks were not completed as efficiently as anticipated, as we observed some technical issues which mostly stemmed from users not following the instructions, such as Task 1 allowing users to move past the target without reaching it if they turned themselves completely away from the target area (and ignored the instructions) and Task 2 not enforcing the use of backwards movement, which is why some users who did not read or understand the instructions or instruction video were able to complete the task without moving backwards at all. We also noticed clear preference toward physical turning in comparison to artificial turning, as in Task 5, users often turned their head at the same time when turning with the "pistol" gestures, and in Task 12, most users opted to not use the artificial rotation option at all. This leads us to ponder whether artificial self-orienting mechanics are necessary at all in VR applications, as they clearly are not natural when comparing how we orient ourselves in the real world. Then again, the play area limitations and attached cable management for the VR headsets are often justified reasons to design this option into the software. Perhaps in the future, all VR headsets will become wireless similar to the Meta Quest 2, and artificial turning can become an optional design niche for those with physical disabilities involving rotational self-movement and/or other physical movement.

We also received further confirmation about the lesser usability of middle-finger pinch when compared to index-finger pinch, as in Task 6, almost all users initially teleported with the index-finger pinch and only after re-reading the instructions realized the possibility of using the middle-finger pinch as well. The index-finger pinching worked quite well overall, while attempting to middle-finger pinch was registered as the index-finger pinch a few times for most users. The index-finger pinch was dominantly utilized in teleportation in later tasks as well. Additionally, in many parts of the testing, there was accidental teleportation. This caused many users to teleport away from their objectives, especially high up in the testing scene, thus prompting the need for using the downwards movement actions. There were also some users who moved in the space quite a lot, sometimes moving parts of their hands outside of the view of the recording cameras (and were told to reverse).

For other design improvements, the instructions for how to enable and disable the menu in Task 8 disappeared a bit too quickly in case the task was completed almost instantly, and some test users did not learn how to close or open the menu properly due to this, as they initially did it accidentally. Task 9 worked quite well for some users, but many had issues with the menu popping up or closing due to accidental fist closing or a hand being registered as a fist when index-finger pinching. In general, the menu popped up too easily unintentionally, especially in the one-handed UI, despite this test attempting to eliminate this behavior, which was very apparent in our previous test [3]. However, it is safe to say that switching the forward movement from thumb up to "flathand" did reduce the unwanted behavior considerably.

There were also issues with Task 10, as many users did not realize where the model that they spawned into the scene was (or even what it looked like). This was likely due to not enough cues provided in Task 8 when the model was spawned and poor instruction reading. After finding the model, it was difficult to rotate it correctly, as the info text that told about the remaining rotation amount was often obscured by the model or was too

far away to be read or noticed by the user. Furthermore, in Task 11, the model was often rotated using one hand only, meaning if the users over-rotated it, then instead of switching the hand (which would change the direction), they preferred to rotate the model another 360 degrees to obtain the correct rotation. This behavior indicates uncertainty with the users' ability to re-initiate the gesture. In the future, it would be interesting to see how this behavior would or would not be replicated in a longer study. It could also be wise to enable advanced features, such as omnidirectional rotation for the models, only after the user has become sufficiently capable of using the basic features.

Overall, the 2H UI was clearly more reliable than the 1H UI as intended, showing less involuntary action activation, which leads us to believe that bimanual coordination could be a better option for tasks which do not require quick action sequences but rather precision, such as medical operations or operating potentially dangerous machinery with gesture controls. Then again, unimanual coordination could suit often repeated simple actions better, such as artificial movement. Thus, we believe that the bimanual approach suits complex tasks, while unimanual is the better fit for simple tasks.

When utilizing less well-known gesture controls, their naming conventions can become difficult for the test users to understand, e.g., "fingers forward facing vertical flat hand", and that is why written instructions should always come with visual representations of the gestures (as in our test). The written instructions have other disadvantages as well, as with the English version, some test users had difficulties understanding certain basic words, such as "collide", which prompted them to ask the testing supervisor. The language barriers can be assumed to have slowed down the progress of some users as they either took longer to read the instructions or misunderstood them. Some users also read the instructions out loud (to aid their understanding most likely). Reading and understanding instructions was at a very low level overall; many users did not read/understand them properly and attempted incorrect actions constantly until prompted to re-read the instructions when they became frustrated. This leads us to believe that directions should be in even more visual form than now when introducing users to gesture UIs. Perhaps rather than only an instruction video, the application could provide a 3D representation of the gesture actions near the user for easy comparisons with their own hands and refrain from attempting to describe the gestures in plain text or spoken words.

However, the issues during the test were not the only part that reduced the reliability of our results, as the questionnaire part also suffered from poor instruction-reading and language barriers. Additionally, some users clearly had some cultural or personal reasons for not wanting to ask when they did not understand something. This was shown in a prominent way when some users were asked which version they thought they had chosen as preferred in the final question and to describe its mechanics. We found at least three users who ended up reversing their choice, but without the question prompt, the version preference would have been selected incorrectly; thus, the result must have been affected by this as well. While this is a large issue as the final choice culminates the entire questionnaire, the other data, in many cases, can be used to see if the choice makes sense.

Finally, a major VR usability-related observation was that as there were many test users with glasses, some had major difficulties fitting them inside the Meta Quest 2 headset, which showcases that the technology is not completely accessible just yet. This further requires the VR designers to consider what type of content they can include in the applications, as small details could be invisible to some users with certain VR hardware.

Our suggestions for the future of gesture UI research include testing similar interfaces with more varied and larger samples, especially samples with a close to equal number of right- and left-handed participants, so that the impact of hand dominance on gesture preference can be understood better. There could also be a different user interface for both hand dominance types. Additionally, it would be beneficial to test whether experience in controller-based UIs in general slows down or speeds up the learning of gesture UIs. Another potential direction would be to mirror the test with standardized questionnaires, such as using the SSQ, SUS, or NASA TLX and compare whether the insights differ.

Furthermore, other disciplines, such as medical science, could utilize gestures in, e.g., remote surgeries or surgical training. However, for those use case scenarios, the systems would have to be much more accurate and robust than the technology is capable of right now due to issues such as occlusion, lighting limitations, network connectivity, and hand shape differences in humans, among others. Additionally, there are currently unsolved issues such as the lack of correct haptic/force feedback which would be able to generate, e.g., the muscle memory required in surgery [73].

7. Conclusions

In this paper, we presented a case study with two hand-tracking (HT) user interfaces: the one-handed (1H UI, unimanual) and the two-handed (2H UI, bimanual). The UIs were developed using the Unity game engine and the Oculus integration software development kit, and the development device was the Meta Quest 2. The idea of the 1H UI is that it is simpler to utilize and requires less coordination between hands with the disadvantage that accidental gesture activation can be more frequent, whereas the 2H UI is focused on reliability and preventing accidental gesture activation and comes with the potential drawback of requiring two-hand coordination and maybe causing more physical strain. The user interfaces are designed for the context of 3D design review software in virtual reality, which draws inspiration from a research project called Sustainable Shipbuilding Concepts, where one topic is to utilize VR as a visualization tool. The idea with hand-tracking and gesture UIs is that they could potentially alleviate the entry barriers of VR usage by reducing strain from holding controllers, improving learning by utilizing easy-to-understand gesture controls rather than button inputs, as well as potentially alleviating VR sickness by offering a control scheme which represents the experienced actions, especially artificial motion (vection), better than a controller.

The underlying methodology is similar to *design science*, which in essence means that we designed and created a research artefact (the user interfaces) to investigate a defined problem, which is the usability of unimanual gestures versus bimanual gestures in VR and 3D design review software with an additional motivation to increase VR adoption in the ship construction industry as a visualization tool. Additionally, the artefact can solve issues related to VR usability in general with the help of gesture user interfaces. The interfaces were pilot tested before the real test.

In the test, a testing sample of 25 users was utilized, who tested the two user interfaces sequentially, filling a testing questionnaire after testing each user interface. The test users were further categorised into 14 categories based on internal traits such as previous VR or HT experience, age, gender, testing order, testing language, and whether they were able to complete the testing. The testing was recorded with three cameras recording both the visuals and audio in order to better understand the way the users utilize their hands for various actions. The test was divided into 13 tasks (0–12), which introduced the mechanics to the test users one by one and eventually required the users to apply their learnt skills. Each task contained written instructions and an instruction video within the virtual environment. The tasks were timed, and alongside the task completion times, the amount and duration of gesture and action activations performed by the users were logged as text files for performance-based data analysis.

The results showed that the 1H UI was preferred greatly based on the questionnaire results, and the only category of users that preferred the 2H UI was the category which tested the 1H UI first. This means that all the other traits—age, gender, previous VR and HT experience, and language—did not impact the version preference seemingly much, meaning that the 1H UI was clearly more enjoyable to utilize for the testing sample. This also means that it can be determined that the testing order was a large factor in determining the testing experience, and namely that the experience gained from already having tested one version greatly increased the performance in the second version independent of the utilized user interface or user traits. The result was also statistically significant.

In relation to hand preference, there was a preference for right-hand usage in the 1H UI, while the 2H UI was more varied. As most (24/25) users were right-handed, this could indicate dominant hand preference for single-hand gestures.

When rating the gestures, the fingers forward facing “flathand” gesture was rated clearly the most enjoyable to form, whereas the worst gesture was the “rock-climbing” gesture, as it was rated most often as one of the three disliked gestures by the test users, and it also had the lowest rating of being one of the liked gestures. The “rock-climbing” gesture was also causing the most false positives, as it had a high activation rate even in tasks where it served no purpose. These results indicate that the design of the “rock-climbing” gesture could be iterated upon further, while the “flathand” at least seems suitable for further testing.

The user movement actions were considered easier to utilize than controlling a 3D model or other menu-related actions, although most users had difficulties learning the backwards movement in Task 2, and thus it was barely utilized throughout the test. Users were also utilizing the movement actions at normal speed for the most part. Moreover, when utilizing the pinch gestures, the index-finger pinch was greatly (75%) favored against the middle-finger pinch. Additionally, physically turning in the space was preferred over using the provided artificial turning method with the “pistol” gesture, indicating that physical turning in VR is a more natural way.

The task completion times showed that the 1H UI was faster to complete on average, although the 2H UI was faster when completed as second, but the difference between completing it as first compared with completing it second was more than double, whereas for the 1H UI, the difference was a lot smaller. This indicates that the 2H UI may have a steeper learning curve, but that it is ultimately more efficient than the 1H UI once the users are familiar with it.

The experienced strain and VR sickness results showed that according to the questionnaire, most users experienced none or slight VR sickness or strain, although four users became so nauseous that they had to abort the testing. What is more, one of the test users who had to discontinue ended up vomiting in the end. This result shows that hand-tracking and gesture UIs do not at least completely remove the VR sickness.

Overall, the test suggests that a unimanual control scheme could be preferable to a bimanual one, mainly due to it being simpler and faster to utilize, although a long-term use study would be needed to confirm this result. Such a study could integrate some relevant real-world tasks into the user interfaces and record the results during weeks of usage, comparing the user experience at different stages of user adoption such as novice user, intermediate user, and expert user.

Author Contributions: Conceptualization, T.N., T.L. and J.S.; methodology, T.N.; software, T.N.; validation, T.N., T.L., J.S. and S.H.; formal analysis, T.N.; investigation, T.N.; resources, T.N.; data curation, T.N.; writing—original draft preparation, T.N.; writing—review and editing, T.N., T.L., J.S. and S.H.; visualisation, T.N.; supervision, T.L. and J.S.; project administration, T.L.; funding acquisition, T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Business Finland. The APC was funded by University of Turku.

Institutional Review Board Statement: Ethical review and approval were waived for this study due to no minors being involved and no personal sensitive information collected based on national research ethics guidelines. Additionally, the potential harm to test subjects is minimal and the article to be published contains no identifiable data of the test subjects.

Informed Consent Statement: Written informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data collected during the study are available upon request.

Acknowledgments: The research reported in this article has been conducted as a part of *Sustainable Shipbuilding Concepts (SusCon)* project. The project is carried out in collaboration with VTT Technical

Research Centre of Finland, Evac, Lautex, Meriteollisuus, Meyer Turku, NIT Naval Interior Team, Paattimaakarit, Piikkio Works, and Royal Caribbean Group.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A. Detailed Task Descriptions and Task-Based Results

Appendix A.1. Introduction Task: Instructions (Task 0)

After fitting and adjusting the VR headset, the test started with Task 0 requiring the user to read instructions located on the virtual test environment walls around them. After the user was ready, they needed to close their fists three times simultaneously to start the test. The fist was activated 11–14 times and held closed 13–17 s in both versions on average (1H: Left hand: 5.5 times for 5.6 s, Right hand: 8 times for 11.1 s; 2H: Left hand: 6.0 times for 7.0 s, Right hand: 5.6 times for 6.3 s). This shows that there were perhaps some difficulties in activating the fist gestures simultaneously, especially in 1H UI using the right hand. For the other gestures (see Section 3 for gesture descriptions), the “rock-climbing” gesture was activated rather many times unintentionally (1H: 24 times for 19.9 s; 2H: 21.9 times for 21.4 s).

Appendix A.2. Forward and Backwards Movement (Tasks 1–2)

The goal was to move forward using the “flathand” gesture(s) and cross a 10 m line on the x or z axis, which were drawn in the environment (previously showcased in Figure 4). The optimal solution required one activation with an active time of less than 5 s. The logs show that for normal speed, the 1H UI had 8.8 activations with 3.6 s active time, and 2H UI had 5.5 activations with 4.2 s active time, which was considerably more than double speed (1H: 3.4 times for 1.4 s; 2H: 4.6 times for 1.9 s). This indicates that forward movement had some issues learning the gesture activation, but once the gesture was active, the users moved to the correct direction and did not waste time.

The “flathand” gesture was activated around the same amount of times in both versions (1H: 8.3 times; 2H: 9.2 times), but the active time difference was drastic with 2H UI having double the active time (39.8 s) compared to 1H UI (18.8 s). A cause for this is likely due to the users in the 2H UI not realizing they needed to form the closed fist or thumb up simultaneously with the “flathand”. The relative data showed that the “rock-climbing” gesture was activated many times (around 40%) and for a rather long duration (around 20%) relative to the main gesture “flathand” (around 25% activations and 45% duration), further highlighting that the “rock-climbing” gesture is formed unintentionally rather easily.

The users needed to move back to the origin (starting point) by reversing with the “flathand” gesture and fingers pointing upwards in Task 2. Backwards movement was activated 19.1 times for 6.7 s in the 1H UI and 7.1 times for 6.1 s in the 2H UI at normal speed. For double speed, the statistics were 1H = 11.3 times and 7.3 s; 2H = 4.7 times and 2.3 s. The optimal performance would have been one to two activations for around 5 s or less, but it can be seen that there were considerably more activation times in both UIs, although there were a lot more in the 1H UI. This indicates that learning backwards movement was difficult, and the observations strongly agree with this conclusion.

The relative data show that the “flathand” gesture was active for the longest (as desired), while the “rock-climbing” gesture again had many unnecessary activations. Action activations, as visualized in Table A1, show that forward movement was active for around the same duration as backwards movement. This indicates that users maybe did not realize they needed to stop the forward movement when reaching the 10 m line on the floor (observed behavior) in Task 1; thus, the duration “overflowed” to Task 2, and also that some users had difficulties in reversing (observed behavior) and ended up manually turning their head and moving forward instead. Additionally, there were many observations about users not understanding the goal of the task clearly, as the word “origin”, or even the

phrase “starting point” was not understood correctly by some users. Thus, some users did not realize where the origin was in the environment.

Table A1. Relative action usage in Task 2, calculated with the relative calculation mechanism described in Section 4.3. The highest value(s) for each version in both action speeds is highlighted in red, while the lowest value(s) is highlighted in blue.

Action	Relative Activations				Relative Active Time			
	Normal Speed		Double Speed		Normal Speed		Double Speed	
	1H	2H	1H	2H	1H	2H	1H	2H
FORW	30%	26%	16%	25%	31%	28%	18%	17%
BACK	38%	29%	17%	20%	30%	39%	20%	17%

Appendix A.3. Up and Down Movement (Tasks 3–4)

Task 3 was to learn up movement using the “flathand” gesture with the palm facing upwards, and the goal was to rise up and reach the 10 m line on the y axis in the virtual environment. The up movement was activated 2.9 times for 2.1 s in the 1H UI and 2.4 times for 2.5 s in the 2H UI at normal speed, which was more than double speed (1H: 0.8 times for 0.8 s; 2H: 2.0 times for 0.6 s) similar to the earlier task results. Both versions were performed close to optimally, so there probably were no major issues with this task.

The closed fist was over-utilized in the 1H UI, which could indicate difficulties with the task completion. Both versions activated the “rock-climbing” gesture unnecessarily, as in previous tasks. Additionally, action activation statistics show that the upwards movement was preferred for both versions, with normal speed over double speed.

In Task 4, users were required to descend to the ground from their previous height (of 10 m optimally). The gesture to learn was the “flathand” with palm facing downwards with similar optimal activation estimates than prior tasks. The activations were 3.1 times for 1.6 s in the 1H UI and 2.1 times for 2.8 s in the 2H UI at normal speed. For double speed, usage was 1.6 times for 1.8 s in the 1H UI and 1.6 times for 1.0 s in the 2H UI, showing little difference to normal speed. Neither version seemed to face major issues with learning the action.

The gesture-forming statistics mirror the findings in previous tasks about the prevalence of the “rock-climbing” gesture. Action activations statistics show that the downwards movement was clearly the most utilized action, and additionally that in the 1H UI, users preferred to move down at double speed, while in the 2H UI, the normal speed was used more. Additionally, some upwards movement carried over from Task 3 was observed.

Appendix A.4. Turning the User (Task 5)

Task 5 introduced an artificial smooth rotation option, which afforded the user to turn left or right using the “pistol” gesture. The goal was to turn 360 degrees in either direction combined, with the optimal performance being around one to three activations for the “pistol” for 7.5–10 s. The results show that users turned 8.4 times for 13.8 s in the 1H UI and 4.9 times for 8.5 s in the 2H UI at normal speed. The double speed turning (2H only) was 3.1 times for 1.5 s. Users turned for longer in the 1H UI, but this is easily explained by the lack of double speed option, which caused them to take longer to complete the turning. Overall, it seems there were some issues in the “pistol” gesture recognition due to there being a large activation amount.

The “pistol” gesture was not activated in such relative amounts as it should have been. This is especially true for 1H UI, as it had a large unnecessary allocation for the closed-fist gesture. Additionally, the “rock-climbing” gesture saw less usage than in prior tasks, which was an indication that users might be learning to better position their hands for the actions that are needed or to avoid the easily occurring accidental recognition. For pinch gestures, the amount of no pinching at all went considerably down from the previous

tasks, indicating that pinching might occur rather easily unintentionally when attempting to form the “pistol” gesture.

Turning was overall the most used action, and there is not much difference in the relative amounts when the fact that 2H UI allows for double-speed turning is taken into account. The 1H UI had a rather large allocation for the forward movement, which is strange, as the task is optimally completed staying still, and this was not replicated in the 2H UI. One potential explanation is that users had some issues with their “pistol” being recognized as a forward-facing “flathand”, which activated the forward movement. The reason why this did not occur in the 2H UI could be because of the dual hand requirement.

Appendix A.5. Teleportation (Task 6)

Task 6 featured learning how to teleport using either the index-finger pinch (INDTELE) or the middle-finger pinch (MIDTELE). First, the users needed to activate a ray in their left hand using the “rock-climbing” gesture and aim it at a teleportable surface (such as the floor or air), which subsequently made the ray’s color turn green (from red). Second, the users needed to utilize either one of the pinch gestures to teleport to the ray’s end location. The goal of the task was to teleport three times using both pinch methods; thus, the optimal solution was six teleports in total with similar ray activations. There were 4.6 INDTELEs and 3.6 MIDTELEs in the 1H UI, while in the 2H UI, the result was 6.8 INDTELEs and 3.3 MIDTELEs. This shows that there potentially were difficulties in activating MIDTELE in the 2H UI, as the high amount of INDTELEs probably means that an attempted MIDTELE was recognized as an INDTELE multiple times, which was also noted as an observation for some users. It is unclear why this happened in the 2H UI only, as the mechanic was identical in both. For ray activation, the 2H UI had more active time (1 min and 4.5 s) and activations (32.6) than the 1H UI (43.5 s and 28.5 activations), which is also a sign of difficulties. In addition, only the left hand was capable of initiating a teleport; however, it was observed that some users did not properly read the instructions and instead attempted to teleport with the right hand, sometimes taking tens of seconds to realize their error.

For gesture activation, the “rock-climbing” gesture was clearly activated and stayed active the most, although there was a rather large percentage of both closed fist and “flathand” gestures, even though they were unnecessary. One reason for this can be that users positioned themselves to see the instructions better after initiating a teleport. Another is that some users were observed to (accidentally) teleport quite high up and subsequently wanted to quickly come back down due to VR-induced acrophobia while simultaneously being unable to control the teleportation well enough to descend using it, thus relying on the downwards movement action. It can also be seen that the index-finger pinch was used for around two-thirds of the activations for both versions, which corresponds to the teleport type statistics.

Appendix A.6. Positioning Oneself in the Virtual Environment (Task 7)

In Task 7, the users were required to move to a specific position in the virtual environment, which contained a yellow box-like object which the users were told to collide with using any movement mechanics they chose. This task had no defined way of completion; thus, there is no optimal action amount.

The action activations in Table A3 show that forward movement was majorly utilized in both versions both at normal and double speed. There was also a relevant amount of downwards movement; this is likely because the users often teleported quite high up in Task 6, so now they needed to descend to the objective. Some amount of turning was also utilized, but there was not very much backwards or upwards movement, the reasons for which are probably that many users did not properly learn backwards movement and there was no need to ascend in the task for the most part. When comparing the action activations to the gesture activations in Table A2, the “rock-climbing” gesture emerged on top, while “flathand” and closed fist were close in comparison. Additionally, the 1H UI utilized a relatively large amount of closed-fist gestures despite it being useless. This likely means

gesture recognition issues. Finally, the amount of teleportation used in the task was on a similar level to the previous task (1H: 3.6 INDTELEs, 2.2 MIDTELEs; 2H: 2.9 INDTELEs, 2.0 MIDTELEs) with a slight preference for INDTELE.

Table A2. Relative gesture usage in Task 7, calculated with the relative calculation mechanism described in Section 4.3. The highest value(s) for each version in both gesture categories is highlighted in red, while the lowest value(s) is highlighted in blue.

Non-Pinch Gestures	Relative Activations		Relative Active Time	
	1H	2H	1H	2H
Closed fist gesture	17%	24%	22%	30%
“Flathand” gesture	32%	19%	39%	24%
“Rock-climbing” gesture	43%	32%	35%	35%
“Pistol” gesture	9%	3%	4%	3%
Thumb-up gesture	-	22%	-	8%
Pinch gestures	1H	2H		
Index-finger pinch	68%	64%		
Middle-finger pinch	18%	31%		
No pinches	14%	5%		

Table A3. Relative action usage in Task 7, calculated with the relative calculation mechanism described in Section 4.3. The highest value(s) for each version in both action speeds is highlighted in red, while the lowest value(s) is highlighted in blue.

Action	Relative Activations				Relative Active Time			
	Normal Speed		Double Speed		Normal Speed		Double Speed	
	1H	2H	1H	2H	1H	2H	1H	2H
FORW	35%	29%	11%	15%	33%	30%	11%	11%
BACK	7%	3%	1%	2%	5%	3%	1%	2%
UP	5%	6%	2%	5%	3%	7%	3%	5%
DOWN	23%	15%	5%	7%	19%	18%	12%	7%
TURNRL	12%	6%	-	3%	14%	7%	-	2%

When looking at the hand preferences, the 1H UI had no clear preference toward either hand, while the 2H UI had a considerably higher preference rate toward left-hand usage, especially in the actions that were activated the most. The latter result is also replicated via the absolute gesture activation statistics, showing that the right hand was extensively used for the fist and thumb up to control speed, but it was used a lot less on the other gestures to control action type. Additionally, the 1H UI had the option of combining actions, but most users did not utilize it.

Appendix A.7. Menu Activation and Placement (Task 8)

The menu, displayed in Figure A1, could be made to appear by either closing the left fist (1H UI) or both fists simultaneously (2H UI), and it followed the position and orientation of the user’s left wrist/fist while the left fist was held closed (for both versions). Opening the left fist made the menu stay in its current position, which was the intended way to interact with the menu without having to hold the left hand stable. The menu could be interacted with by activating the ray with the right hand and pointing it at the menu buttons (which made the ray turn blue), which could then be selected by an index-finger pinch with the same hand. Additionally, the menu could be closed (made invisible) by closing the right fist in the 1H UI (when the left fist was not simultaneously closed) or by

closing both fists again (first both need to be opened) in the 2H UI. The task goal was to learn menu usage basics by opening it and first selecting a *Choose Object* button and then choosing one of the four possible models to spawn into the scene (to the origin) by selecting a button with the model's name. The models can be seen in Figure A2.

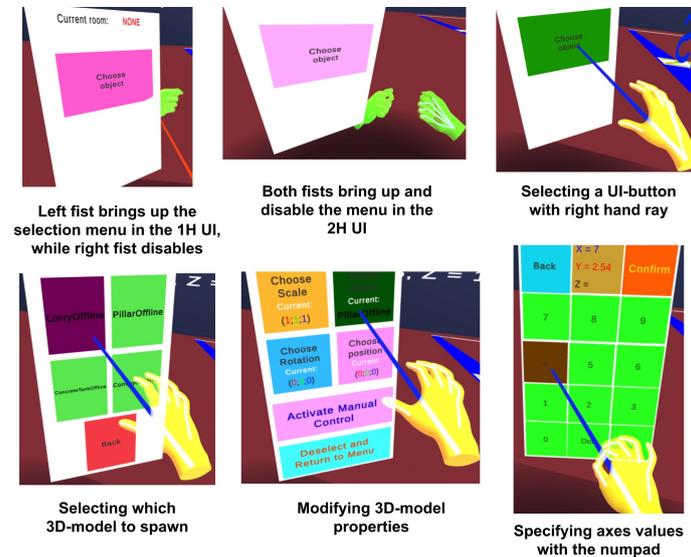


Figure A1. The menu that is used to spawn 3D models into the scene and modify their properties. Colours exist to differentiate sections of the menu, while the numpad is used to enter coordinates for object moving.

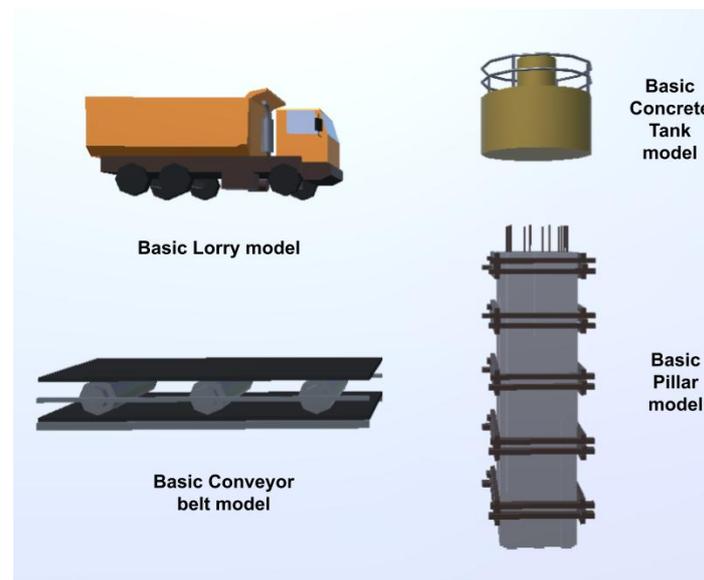


Figure A2. The four models available to be spawned into the scene in Task 8 (or later), which are a basic lorry, conveyor belt, concrete tank, and pillar.

The only difference in the task between the versions was the menu opening and closing mechanic. The required gestures were the fist(s), right-hand “rock-climbing”, and index-finger pinch. The optimal solution involved one to two menu openings and two index-finger pinches, with one to two fist(s) activations, while the menu should remain active for a maximum of 10 s. The menu was activated on average five times for 1 min and 7.5 s in the 1H UI, and 2.3 times for 32.0 s in the 2H UI. The closed fist was also activated considerably more in the 1H UI (33.0 times for 77.3 s) compared to the 2H UI (17.6 times for 42.3 s). Overall, both results were far from the optimal in terms of menu active time,

which indicates that learning the basic menu functions was difficult. It was also observed that many users did not realize they needed to leave the menu stationary, which impacted their ease of use.

The activation of the menu buttons was logged, and the results show that the *Choose Object* button required in the task was selected 1.2 times in the 1H UI and 1.7 times in the 2H UI, showing that some users reversed their selection (to be able to choose it more than once), indicating poor task understanding or aiming inaccuracy. Out of the four options, the Lorry model was chosen to be spawned clearly the most, although this is understandable, as it was featured in the instruction video. Furthermore, the 2H UI was completed a lot faster, which was mainly due to the right hand accidentally disabling the menu when attempting to pinch and select buttons with it in the 1H UI, as it was recognized as a closed fist.

Appendix A.8. Understanding the Menu and Selecting Buttons (Task 9)

In Task 9, the user needed to navigate the menu by selecting *Position Select*, *Custom Position*, and then setting the positional coordinates for their spawned object to $x = 10$, $y = 5$, and $z = 10$. Completing these actions moved the model from the origin to those coordinates instantaneously. There were also unnecessary buttons available to see how accurately the users were able to select the correct buttons using the ray interaction. The optimal solution was to select the *Position Select* once, then *Custom Position*, and then from the numpad: *1*, *0*, *Confirm*, *5*, *Confirm*, *1*, *0*, *Confirm*. Thus, the deviation from the aforementioned button presses shows the rate of error or misunderstanding. The most selected buttons in this task were the ones required for completion for both versions. Additionally, the *Numpad Delete* was utilized around once per test user on average, which signals a small amount of wrong inputs. Some unnecessary button usage was noticed with the rotation selection options and *Numpad 8* in addition to returning from the numpad selection in the 1H UI. However, the 2H UI showed barely any inputs on the wrong numbers or other than position-related options; thus, this could mean that it was easier for the users to stabilize the menu in the 2H UI, leading to fewer input errors.

Appendix A.9. Manual Model Rotation (Task 10)

In Task 10, the objective was to rotate the model around the x and z axes in world space, 360 degrees on both. In order to accomplish this, the user needed to activate the manual control for the object from the menu using the *Activate manual control* button. When the manual control was enabled, the user's movement actions controlled the model instead of the user (apart from the teleport), and additionally, the model could be rotated on all three axes, while the user can only turn on the y axis. The model turning for the task functioned by pointing the "pistol" gesture either upwards or downwards, which determined the direction of the rotation on the axis. The rotated axis was determined by the hand which formed the gesture; the left hand rotated on the z axis, while the right rotated on the x axis. Similar to Task 5, the 2H UI had the opportunity to rotate the model at double speed as well as normal, while the 1H UI only had the normal speed option.

Many users had issues with either understanding how to enable the manual control or realizing that they needed to rotate the model on both axes. This issue was intensified by the users being too far away from the model to see the text counter attached to it, which showed the remaining rotation amounts. There were also issues with the menu activating when trying to rotate the model as well as users not pointing their "pistol" gestures at high or low enough angles and thus ending up rotating on the y axis for an extended period of time. Many users also had difficulties in finishing reading the task instructions. The optimal solution involved one to two activations for either turning action, while the rotation itself should take around 15–20 s.

For z axis turning, there were 14.8 activations for 15.3 s in the 1H UI and 15.5 activations for 17.1 s in the 2H UI. Respectively, the x turning results were 20.2 activations for 19.9 s and 10.3 activations for 15.4 s. The 2H UI additionally had the double-speed option: 13.3 activations for 9.5 s z turning and 4.9 activations for 2.5 s x turning. The statistics show

that overall, z turning (left hand) was slightly favored in the 2H UI, although considerably so when using double speed, while the x turning (right hand) was favored in the 1H UI. The result is consistent with previous findings which suggest that in the 1H UI, the right hand was preferred, while in the 2H UI, the left hand was utilized more. Furthermore, when inspecting the rotation direction, the negative direction was clearly favored in both versions, indicating that pointing the “pistol” gesture upwards would be preferable to pointing it downwards.

Appendix A.10. Manual Model Positioning (Task 11)

In this task, the users needed to move the model into a shape container in such a way that it was overlapping a green cube within the container without touching the container’s walls (see Figure A3a). This task afforded using any of the movement controls, although the model cannot be teleported. The optimal amount of action usage is difficult to estimate, but considering the distance that the model should be moved based on its intended location after the previous task, the total movement amount should be around 15 units (meters) along the x axis (from coordinate (10, 5, 10) to (25, 5, 10)). This movement amount should require at most 10–15 s, but the difficulty lies in positioning the model within the shape container, and this could take an extensive period of time based on user skill and spatial understanding.

According to the observations, many users had issues with the rotation controls, which improved only negligibly from Task 10. Users were also faced with standard object position controlling issues, such as moving too far forward or back and missing the target, in addition to issues with the menu being activated unintentionally, thus disabling the movement controls until it was deactivated again. Furthermore, some users did not realize that the menu had opened in cases when they accidentally activated it outside of their field of view and thus spent a long time frustrated when nothing was functioning. This can be seen by the absolute menu activations numbers, which were 21.5 activations for 1 min and 31.5 s in the 1H UI compared to 16.9 activations and 1 min and 57.3 s in the 2H UI. As the menu was only required to be activated for enabling the manual control, any activation amount greater than one suggests that the users had issues with uninitiated menu activation or disabling or both.

Table A4. Relative gesture usage in Task 11, calculated with the relative calculation mechanism described in Section 4.3. The highest value(s) for each version in both gesture categories is highlighted in red, while the lowest value(s) is highlighted in blue.

Non-Pinch Gestures	Relative Activations		Relative Active Time	
	1H	2H	1H	2H
Closed fist gesture	22%	22%	23%	33%
“Flathand” gesture	21%	16%	24%	19%
“Rock-climbing” gesture	40%	30%	33%	23%
“Pistol” gesture	18%	9%	20%	15%
Thumb-up gesture	-	24%	-	10%
Pinch gestures	1H	2H		
Index-finger pinch	74%	81%		
Middle-finger pinch	26%	19%		

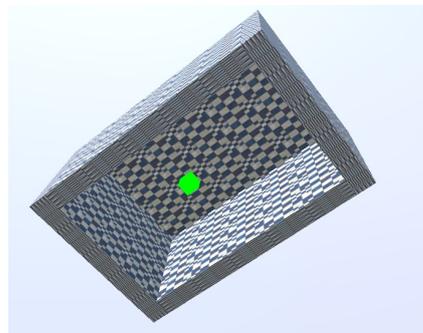
The relative gesture data in Table A4 show that the “rock-climbing” gesture for ray activation was used the most in general, although the fist usage was slightly higher in duration in the 2H UI, which is understandable considering the requirement for any non-pinch action to activate. When looking at the teleportation data to correlate with the high “rock-climbing” usage, the result shows that there were around 10 teleports in total

for both versions, indicating that the ray usage was mostly related to utilizing the menu or unintentional.

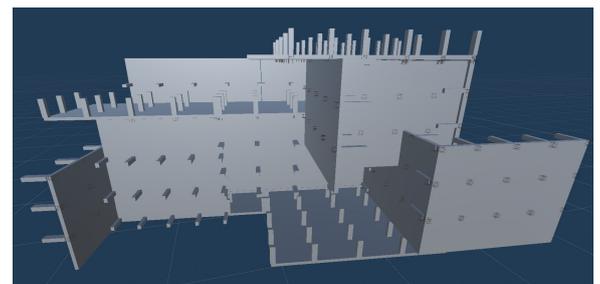
Table A5. Relative model action usage in Task 11, calculated with the relative calculation mechanism described in Section 4.3. The highest value(s) for each version in both action speeds is highlighted in red, while the lowest value(s) is highlighted in blue.

Action	Relative Activations				Relative Active Time			
	Normal Speed		Double Speed		Normal Speed		Double Speed	
	1H	2H	1H	2H	1H	2H	1H	2H
FORW	25%	22%	4%	11%	18%	20%	5%	6%
BACK	3%	3%	0%	1%	2%	2%	0%	1%
UP	3%	3%	1%	2%	2%	3%	1%	1%
DOWN	8%	6%	1%	3%	5%	4%	2%	2%
TURNRL	31%	18%	-	12%	42%	31%	-	11%
TURNUPMOD	15%	10%	-	5%	18%	13%	-	3%
TURNDOWNMOD	7%	2%	-	2%	5%	2%	-	1%

The relative model action data in Table A5 show that the model was moved the most using forward movement and y turning actions for both versions, while for x and z turning, the negative direction was preferred over the positive direction, which was similar to Task 10. Furthermore, similar to the user movement actions, the downwards movement was utilized considerably more than the upwards movement, which could mean that perceiving the movement was easier downwards or that some reason caused the model to move up unintentionally, such as rotating the model so that its forward direction became up.



(a) The shape container for Task 11 that the user needs to direct their model into. The model must touch the green cube inside the container without touching the container walls, which prevents the model being inserted only partially or overlapping.



(b) The building-sized model which spawns into the scene at the start of Task 12. The test user needs to move inside it and find a hidden red box. Teleporting is not allowed inside the model.

Figure A3. Task 11 and 12 elements.

Appendix A.11. Applying the Learned Skills through Exploration (Task 12)

The users were instructed to navigate inside a large industrial-appearing 3D model and locate a red box within it, as shown in Figure A3b. When a user found the box, they needed to aim the right-hand ray to it (which turned yellow) and index-finger pinch to finish the testing. The large model is made to contain a lot of reference points (in the form of shapes) so that the user can visualize their movement more easily. The model is impenetrable by the user, and its surfaces are non-teleportable (to enforce also other ways

of movement). The idea for the task was to test how well the users can apply the skills they have learned in an exploration task and what actions they choose to utilise the most. A task like this can take a long time because of its exploratory nature, so it should encourage the test users to utilize the easiest movement actions for them, which would give a good idea for what actions are the best out of the set.

The optimal task performance was difficult to determine because of the exploration aspect. The relative gesture activation statistics in Table A6 show that the “rock-climbing” gesture was activated the most times for both versions, which strongly suggests that it was unintentionally activating constantly, as there was little need for it apart from potential teleporting outside the large model. The result can be confirmed by looking at the teleport amount, which was 3.5 for the 1H UI and 2.8 for the 2H UI. The “flathand” gesture was active for the longest duration, which was to be expected as it is used to control four different actions. Additionally, a rather high amount of closed-fist activations and active time in the 1H UI, which has been noticed in previous tasks, was noticed here as well. This is likely related to both unintentional menu activation and users’ hand rest or default poses resembling a fist.

The user movement activation statistics, see Table A7, show that forward movement was vastly preferred and that backwards movement was barely utilized in either version. In general, turning was underutilized despite the task environment containing many obstacles that the users needed to avoid while moving. This was related to the observation that users tended to turn around physically instead of using the gestures during the task. This indicates that artificial turning is not intuitive or natural even when using gesture controls.

Table A6. Relative gesture usage in Task 12, calculated with the relative calculation mechanism described in Section 4.3. The highest value(s) for each version in both gesture categories is highlighted in red, while the lowest value(s) is highlighted in blue.

Non-Pinch Gestures	Relative Activations		Relative Active Time	
	1H	2H	1H	2H
Closed fist gesture	16%	25%	13%	32%
“Flathand” gesture	32%	17%	52%	34%
“Rock-climbing” gesture	45%	31%	32%	21%
“Pistol” gesture	7%	3%	3%	3%
Thumb-up gesture	-	24%	-	10%
Pinch gestures	1H	2H		
Index-finger pinch	84%	79%		
Middle-finger pinch	16%	21%		

Table A7. Relative action usage in Task 12, calculated with the relative calculation mechanism described in Section 4.3. The highest value(s) for each version in both action speeds is highlighted in red, while the lowest value(s) is highlighted in blue.

Action	Relative Activations				Relative Active time			
	Normal Speed		Double Speed		Normal Speed		Double Speed	
	1H	2H	1H	2H	1H	2H	1H	2H
FORW	46%	41%	12%	21%	42%	47%	16%	14%
BACK	3%	1%	1%	1%	2%	1%	0%	1%
UP	10%	9%	3%	6%	10%	12%	6%	5%
DOWN	16%	8%	4%	4%	12%	8%	5%	2%
TURNRL	5%	7%	-	3%	6%	7%	-	2%

References

1. Davila Delgado, J.M.; Oyedele, L.; Beach, T.; Demian, P. Augmented and Virtual Reality in Construction: Drivers and Limitations for Industry Adoption. *J. Constr. Eng. Manag.* **2020**, *146*, 04020079. [\[CrossRef\]](#)
2. Laurell, C.; Sandström, C.; Berthold, A.; Larsson, D. Exploring barriers to adoption of Virtual Reality through Social Media Analytics and Machine Learning—An assessment of technology, network, price and trialability. *J. Bus. Res.* **2019**, *100*, 469–474. [\[CrossRef\]](#)
3. Nyyssönen, T.; Helle, S.; Lehtonen, T.; Smed, J. A Comparison of Gesture and Controller-based User Interfaces for 3D Design Reviews in Virtual Reality. In Proceedings of the 55th Annual Hawaii International Conference on System Sciences, Virtual, 3–7 January 2022; p. 10. Available online: <http://hdl.handle.net/10125/79552> (accessed on 31 January 2022).
4. Buchanan, E.; Loporcaro, G.; Lukosch, S. On the Effectiveness of Using Virtual Reality to View BIM Metadata in Architectural Design Reviews for Healthcare. *Multimodal Technol. Interact.* **2023**, *7*, 60. [\[CrossRef\]](#)
5. Sustainable Shipbuilding Concepts. Research Project. Available online: <https://ar.utu.fi/suscon/> (accessed on 28 July 2021).
6. Wang, J.; Mueller, F.; Bernard, F.; Sorli, S.; Sotnychenko, O.; Qian, N.; Otaduy, M.A.; Casas, D.; Theobalt, C. RGB2Hands: Real-Time Tracking of 3D Hand Interactions from Monocular RGB Video. *ACM Trans. Graph.* **2020**, *39*, 1–16. [\[CrossRef\]](#)
7. Lemak, S.; Chertopolokhov, V.; Uvarov, I.; Kruchinina, A.; Belousova, M.; Borodkin, L.; Mironenko, M. Inertial Sensor Based Solution for Finger Motion Tracking. *Computers* **2020**, *9*, 40. [\[CrossRef\]](#)
8. Gorobets, V.; Merkle, C.; Kunz, A. Pointing, Pairing and Grouping Gesture Recognition in Virtual Reality. In Proceedings of the ICCHP Conference on Computers Helping People with Special Needs, Lecco, Italy, 11–15 July 2022; Lecture Notes in Computer Science; pp. 313–320. [\[CrossRef\]](#)
9. Rakkolainen, I.; Farooq, A.; Kangas, J.; Hakulinen, J.; Rantala, J.; Turunen, M.; Raisamo, R. Technologies for Multimodal Interaction in Extended Reality—A Scoping Review. *Multimodal Technol. Interact.* **2021**, *5*, 81. [\[CrossRef\]](#)
10. Fitzmaurice, G.W.; Ishii, H.; Buxton, W.A.S. Bricks: Laying the Foundations for Graspable User Interfaces. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Denver, CO, USA, 7–11 May 1995; CHI '95; pp. 442–449. [\[CrossRef\]](#)
11. Fitzmaurice, G.W.; Buxton, W. An Empirical Evaluation of Graspable User Interfaces: Towards Specialized, Space-Multiplexed Input. In Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, New York, NY, USA, 22–27 March 1997; CHI '97; pp. 43–50. [\[CrossRef\]](#)
12. Ha, T.; Woo, W. An empirical evaluation of virtual hand techniques for 3D object manipulation in a tangible augmented reality environment. In Proceedings of the 2010 IEEE Symposium on 3D User Interfaces (3DUI), Waltham, MA, USA, 20–21 March 2010; pp. 91–98. [\[CrossRef\]](#)
13. Kang, H.J.; Shin, J.h.; Ponto, K. A Comparative Analysis of 3D User Interaction: How to Move Virtual Objects in Mixed Reality. In Proceedings of the 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Virtual, 27 March–3 April 2020; pp. 275–284. [\[CrossRef\]](#)
14. Bo, Z.H.; Zhang, H.; Yong, J.H.; Gao, H.; Xu, F. DenseAttentionSeg: Segment hands from interacted objects using depth input. *Appl. Soft Comput.* **2020**, *92*, 106297. [\[CrossRef\]](#)
15. Walter, R.; Bailly, G.; Valkanova, N.; Müller, J. Cuenesics: Using Mid-Air Gestures to Select Items on Interactive Public Displays. In Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services, New York, NY, USA, 23 September 2014; MobileHCI '14; pp. 299–308. [\[CrossRef\]](#)
16. Groenewald, C.; Anslow, C.; Islam, J.; Rooney, C.; Passmore, P.; Wong, W. Understanding 3D Mid-Air Hand Gestures with Interactive Surfaces and Displays: A Systematic Literature Review. In Proceedings of the 30th International BCS Human Computer Interaction Conference: Fusion, Swindon, GBR, Poole, UK, 11–15 July 2016; HCI '16. [\[CrossRef\]](#)
17. Remizova, V.; Sand, A.; MacKenzie, I.S.; Špakov, O.; Nyyssönen, K.; Rakkolainen, I.; Kylliäinen, A.; Surakka, V.; Gizatdinova, Y. Mid-Air Gestural Interaction with a Large Fogsreen. *Multimodal Technol. Interact.* **2023**, *7*, 63. [\[CrossRef\]](#)
18. Koutsabasis, P.; Vogiatzidakis, P. Empirical Research in Mid-Air Interaction: A Systematic Review. *Int. J. Human-Computer Interact.* **2019**, *35*, 1747–1768. [\[CrossRef\]](#)
19. Arora, R.; Kazi, R.H.; Kaufman, D.M.; Li, W.; Singh, K. MagicalHands: Mid-Air Hand Gestures for Animating in VR. In Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, New York, NY, USA, 20–23 October 2019; UIST '19, pp. 463–477. [\[CrossRef\]](#)
20. Lystbæk, M.N.; Rosenberg, P.; Pfeuffer, K.; Grønbæk, J.E.; Gellersen, H. Gaze-Hand Alignment: Combining Eye Gaze and Mid-Air Pointing for Interacting with Menus in Augmented Reality. *Proc. ACM Hum.-Comput. Interact.* **2022**, *6*, 1–18. [\[CrossRef\]](#)
21. Reifinger, S.; Laquai, F.; Rigoll, G. Translation and rotation of virtual objects in Augmented Reality: A comparison of interaction devices. In Proceedings of the 2008 IEEE International Conference on Systems, Man and Cybernetics, Singapore, 12–15 October 2008; pp. 2448–2453. [\[CrossRef\]](#)
22. Kangas, J.; Kumar, S.K.; Mehtonen, H.; Järnstedt, J.; Raisamo, R. Trade-Off between Task Accuracy, Task Completion Time and Naturalness for Direct Object Manipulation in Virtual Reality. *Multimodal Technol. Interact.* **2022**, *6*, 6. [\[CrossRef\]](#)
23. Guiard, Y. Asymmetric Division of Labor in Human Skilled Bimanual Action. *J. Mot. Behav.* **1987**, *19*, 486–517. [\[CrossRef\]](#)
24. Cutler, L.D.; Fröhlich, B.; Hanrahan, P. Two-handed direct manipulation on the responsive workbench. In Proceedings of the 1997 Symposium on Interactive 3D Graphics, Providence, RI, USA, 27–30 April 1997; pp. 107–114.

25. Schäfer, A.; Reis, G.; Stricker, D. Controlling Teleportation-Based Locomotion in Virtual Reality with Hand Gestures: A Comparative Evaluation of Two-Handed and One-Handed Techniques. *Electronics* **2021**, *10*, 715. [CrossRef]
26. Radhakrishnan, U.; Koumaditis, K.; Chinello, F. A systematic review of immersive virtual reality for industrial skills training. *Behav. Inf. Technol.* **2021**, *40*, 1310–1339. [CrossRef]
27. Jimeno-Morenilla, A.; Sánchez-Romero, J.L.; Mora-Mora, H.; Coll-Mirallas, R. Using virtual reality for industrial design learning: A methodological proposal. *Behav. Inf. Technol.* **2016**, *35*, 897–906. [CrossRef]
28. Zhang, X.; Jiang, S.; Ordóñez De Pablos, P.; Lytras, M.D.; Sun, Y. How virtual reality affects perceived learning effectiveness: A task–technology fit perspective. *Behav. Inf. Technol.* **2017**, *36*, 548–556. [CrossRef]
29. Challenor, J.; White, D.; Murphy, D. Hand-Controlled User Interfacing for Head-Mounted Augmented Reality Learning Environments. *Multimodal Technol. Interact.* **2023**, *7*, 55. [CrossRef]
30. Al-Shamayleh, A.S.; Ahmad, R.; Abushariah, M.A.M.; Alam, K.A.; Jomhari, N. A systematic literature review on vision based gesture recognition techniques. *Multimed. Tools Appl.* **2018**, *77*, 28121–28184. [CrossRef]
31. Barricelli, B.R.; Gadia, D.; Rizzi, A.; Marini, D.L.R. Semiotics of virtual reality as a communication process. *Behav. Inf. Technol.* **2016**, *35*, 879–896. [CrossRef]
32. Prisacariu, V.A.; Reid, I. Robust 3D hand tracking for human computer interaction. In Proceedings of the 2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG), Santa Barbara, CA, USA, 21–25 March 2011; pp. 368–375. [CrossRef]
33. Measurand Inc. *ShapeHand Data Glove*; 2007.
34. Noitom. Hi5 VR Glove. 2018. Available online: <https://hi5vrglove.com/> (accessed on 8 February 2023).
35. Fredriksson, J.; Ryen, S.B.; Fjeld, M. Real-Time 3D Hand-Computer Interaction: Optimization and Complexity Reduction. In Proceedings of the 5th Nordic Conference on Human-Computer Interaction: Building Bridges, New York, NY, USA, 20–22 October 2008; NordiCHI '08. pp. 133–141. [CrossRef]
36. Wang, R.Y.; Popović, J. Real-time hand-tracking with a color glove. *ACM Trans. Graph. (TOG)* **2009**, *28*, 1–8.
37. Wang, X.; Zhu, Z. Vision-based hand signal recognition in construction: A feasibility study. *Autom. Constr.* **2021**, *125*, 103625. [CrossRef]
38. McMahan, R.; Bowman, D.; Zielinski, D.; Brady, R. Evaluating Display Fidelity and Interaction Fidelity in a Virtual Reality Game. *IEEE Trans. Vis. Comput. Graph.* **2012**, *18*, 626–633. [CrossRef]
39. McMahan, R.P.; Lai, C.; Pal, S.K. Interaction Fidelity: The Uncanny Valley of Virtual Reality Interactions. In *Virtual, Augmented and Mixed Reality*; Lackey, S., Shumaker, R., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 59–70. [CrossRef]
40. Fischer, M.H.; Kornmüller, A.E. Optokinetisch ausgeloste Bewegungswahrnehmung und Optokinetischer Nystagmus. *J. Fürpsychologie Neurol.* **1930**, *41*, 273–308.
41. Bonato, F.; Bubka, A.; Palmisano, S.; Phillip, D.; Moreno, G. Vection Change Exacerbates Simulator Sickness in Virtual Environments. *Presence Teleoperators Virtual Environ.* **2008**, *17*, 283–292. [CrossRef]
42. Vuletic, T.; Duffy, A.; Hay, L.; McTeague, C.; Campbell, G.; Grealy, M. Systematic literature review of hand gestures used in human computer interaction interfaces. *Int. J.-Hum.-Comput. Stud.* **2019**, *129*, 74–94. [CrossRef]
43. Vinayak.; Murugappan, S.; Liu, H.; Ramani, K. Shape-It-Up: Hand gesture based creative expression of 3D shapes using intelligent generalized cylinders. *Comput.-Aided Des.* **2013**, *45*, 277–287, Solid and Physical Modeling 2012. [CrossRef]
44. McNeill, D. *Hand and Mind: What Gestures Reveal About Thought*; University of Chicago press: Berlin, Germany; Boston, MA, USA, 1992. [CrossRef]
45. Pisharady, P.; Saerbeck, M. Recent methods and databases in vision-based hand gesture recognition: A review. *Comput. Vis. Image Underst.* **2015**, *141*, 152–165. [CrossRef]
46. Quek, F.K. Eyes in the interface. *Image Vis. Comput.* **1995**, *13*, 511–525. [CrossRef]
47. Quek, F. The catchment feature model: A device for multimodal fusion and a bridge between signal and sense. *Eurasip J. Appl. Signal Process.* **2004**, *2004*, 1619–1636. [CrossRef]
48. Santos, B.S.; Cardoso, J.; Ferreira, B.Q.; Ferreira, C.; Dias, P. Developing 3D Freehand Gesture-Based Interaction Methods for Virtual Walkthroughs. In *Advances in Human and Social Aspects of Technology*; Advances in Human and Social Aspects of Technology; IGI Global: Hershey, PA, USA, 2016; pp. 52–72. [CrossRef]
49. Rimé, B.; Schiaratura, L. Gesture and speech. In *Fundamentals of Nonverbal Behavior*; Cambridge University Press: Cambridge, UK, 1991; pp. 239–281.
50. McNeill, D. So You Think Gestures Are Nonverbal? *Psychol. Rev.* **1985**, *92*, 350–371. [CrossRef]
51. McNeill, D. *Psycholinguistics: A New Approach*; Harper & Row Publishers: New York, NY, USA, 1987.
52. McNeill, D. Gesture and Communication. In *Encyclopedia of Language & Linguistics*, 2nd ed.; Brown, K., Ed.; Elsevier: Oxford, UK, 2006; pp. 58–66. [CrossRef]
53. Wagner, P.; Malisz, Z.; Kopp, S. Gesture and speech in interaction: An overview. *Speech Commun.* **2014**, *57*, 209–232. [CrossRef]
54. Kendon, A. Some uses of gestures. In *Perspectives on Silence*; Tannen, D., Saville-Troike, M., Eds.; Ablex: Norwood, NJ, USA, 1985; pp. 215–234.
55. Rautaray, S.S.; Agrawal, A. Vision based hand gesture recognition for human computer interaction: A survey. *Artif. Intell. Rev.* **2015**, *43*, 1–54. [CrossRef]
56. Butterworth, B.; Hadar, U. Gesture, Speech, and Computational Stages: A Reply to McNeill. *Psychol. Rev.* **1989**, *96*, 168–174. [CrossRef] [PubMed]

57. Beattie, G.; Shovelton, H. Iconic hand gestures and the predictability of words in context in spontaneous speech. *Br. J. Psychol.* **2000**, *91*, 473–491. [[CrossRef](#)] [[PubMed](#)]
58. Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; Chatterjee, S. A Design Science Research Methodology for Information Systems Research. *J. Manag. Inf. Syst.* **2007**, *24*, 45–77. [[CrossRef](#)]
59. Hevner, A.R.; March, S.T.; Park, J.; Ram, S. Design Science in Information Systems Research. *MIS Q.* **2004**, *28*, 75. [[CrossRef](#)]
60. Järvinen, P. Action Research is Similar to Design Science. *Qual. Quant.* **2007**, *41*, 37–54. [[CrossRef](#)]
61. Hart, S.G.; Staveland, L.E. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*; Hancock, P.A., Meshkati, N., Eds.; Advances in Psychology; North-Holland: Amsterdam, The Netherlands, 1988; Volume 52, pp. 139–183. [[CrossRef](#)]
62. Brooke, J. SUS: A quick and dirty usability scale. *Usability Eval. Ind.* **1995**, 189.
63. Kennedy, R.S.; Lane, N.E.; Berbaum, K.S.; Lilienthal, M.G. Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness. *Int. J. Aviat. Psychol.* **1993**, *3*, 203–220. [[CrossRef](#)]
64. Unity3D Game Engine, Unity Technologies. Available online: <https://unity.com/> (accessed on 1 March 2022).
65. Oculus Integration Software Development Kit for Unity. Unity Asset Store Page. Available online: <https://assetstore.unity.com/packages/tools/integration/oculus-integration-82022> (accessed on 8 February 2023).
66. Sutcliffe, A.G.; Kaur, K.D. Evaluating the usability of virtual reality user interfaces. *Behav. Inf. Technol.* **2000**, *19*, 415–426. [[CrossRef](#)]
67. Student. The Probable Error of a Mean. *Biometrika* **1908**, *6*, 1. [[CrossRef](#)]
68. Shapiro, S.S.; Wilk, M.B. An analysis of variance test for normality (complete samples)†. *Biometrika* **1965**, *52*, 591–611. [[CrossRef](#)]
69. Levene, H. *Robust Tests for Equality of Variance*; Stanford University Press: Stanford, CA, USA, 1960; Volume 2, pp. 278–292.
70. Pearson, K. Mathematical contributions to the theory of evolution—On a form of spurious correlation which may arise when indices are used in the measurement of organs. *Proc. R. Soc. Lond.* **1897**, *60*, 489–498. [[CrossRef](#)]
71. de Souza, C.R. Accord.NET Framework© 2009–2017. Available online: <http://accord-framework.net/> (accessed on 8 April 2023).
72. So, R.H.Y.; Lo, W.T.; Ho, A.T.K. Effects of Navigation Speed on Motion Sickness Caused by an Immersive Virtual Environment. *Hum. Factors* **2001**, *43*, 452–461. [[CrossRef](#)]
73. Laspro, M.; Groysman, L.; Verzella, A.N.; Kimberly, L.L.; Flores, R.L. The Use of Virtual Reality in Surgical Training: Implications for Education, Patient Safety, and Global Health Equity. *Surgeries* **2023**, *4*, 635–646. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.