

Directions for 3D User Interface Research from Consumer VR Games

Anthony Steed *Member, IEEE*, Tuukka M. Takala, Daniel Archer, Wallace Lages
and Robert W. Lindeman *Senior Member, IEEE*

Abstract—

With the continuing development of affordable immersive virtual reality (VR) systems, there is now a growing market for consumer content. The current form of consumer systems is not dissimilar to the lab-based VR systems of the past 30 years: the primary input mechanism is a head-tracked display and one or two tracked hands with buttons and joysticks on hand-held controllers. Over those 30 years, a very diverse academic literature has emerged that covers design and ergonomics of 3D user interfaces (3DUIs). However, the growing consumer market has engaged a very broad range of creatives that have built a very diverse set of designs. Sometimes these designs adopt findings from the academic literature, but other times they experiment with completely novel or counter-intuitive mechanisms. In this paper and its online adjunct, we report on novel 3DUI design patterns that are interesting from both design and research perspectives: they are highly novel, potentially broadly re-usable and/or suggest interesting avenues for evaluation. The supplemental material, which is a living document, is a crowd-sourced repository of interesting patterns. This paper is a curated snapshot of those patterns that were considered to be the most fruitful for further elaboration.

Index Terms—Virtual reality, 3D user interfaces, games, interaction patterns, consumer head-mounted displays

1 INTRODUCTION

In the 2010s, the virtual reality (VR) market moved at remarkable speed from a high-end professional and academic market to a consumer market. At the time of writing, there is still a healthy rate of new devices coming to market. There is a move towards cheaper devices that provide full immersion, as well as more expensive devices that push out the fidelity limits. Alongside the consumer devices, there is a plethora of consumer applications: many games, dramas, documentaries, creative applications, health applications and so on. There are now thousands of applications that can be experienced alongside millions of panoramic videos.

We note that the market for VR devices did involve a temporary technical step backwards. Early high-end immersive VR systems, such as the VPL Research RB2, Division Provision 100 and Virtuality 1000CS, provided six-degree-of-freedom (6DOF, 3 degrees of displacement and 3 degrees of rotation) tracking for the head and 6DOF tracking for one or two hand-held controllers. The early consumer market started with devices supporting only 3DOF head tracking (e.g., Oculus DK1, Google Cardboard) and no hand controllers. Today, the consumer market covers devices with a wide range of configurations which has implications we will cover later. However, on the flip side, aside from cost, the current range of devices is light, sometimes untethered and more easily deployable than those previous lab-only systems. Now, more systems are moving beyond hand-held controllers to bare-hand interaction, a capability that usually could only be prototyped with encumbering devices in labs in the 1990s and 2000s.

This creates a situation where 3D user interface (3DUI) research that

might be applicable to modern consumer VR systems stretches back to the early 1990s. This span of time creates challenges for current researchers and developers who are looking for rules of thumb, design guidelines or specific techniques to use with consumer VR systems. 3DUI research can also be found in a variety of academic and industrial venues. There is a separate problem of applying research results from older systems that had, for example, lower screen resolutions or higher latencies, to modern systems; sometimes it is not clear if results generalise and/or whether experiments need to be re-run to inform design decisions. All of this has led to some re-invention of the wheel in modern systems. However, there are underlying principles that still shine through, such as the classic separation of selection, manipulation, locomotion, system control and symbolic input [1], the role of the body of the user in the system [2, 3], reality-based interaction paradigms [4] and design processes specific for immersive systems [5].

In this paper, we take a look at the current range of designs for experiences in consumer VR and ask what generalisable techniques might be identified that expand the repertoire of 3DUI. We are looking for design patterns that might be quite general purpose or at least applicable to a large subset of systems, and that push 3DUI in new directions.

The first aim is to identify patterns that deserve further study outside of a specific design context. That is, they might be generalisable, but we need a further elaboration of the positive and negative attributes. The second aim is to foster collaboration between research labs and designers, by highlighting the tensions between user research and design exploration. For example, as the user community grows, we might need to establish better guidelines for designers, highlight best practices and support greater accessibility.

Because there are now many thousands of experiences available for consumer VR, our approach has not been to systematically review all experiences or to sample popular experiences. We acknowledge that novelty might be found in smaller titles or even demonstrations. Thus, we have crowd-sourced suggestions from a variety of email lists, Slack channels and other social media.

This paper is not an attempt to generate guidelines for future designers nor a rule book to follow. It simply highlights some areas of 3DUI design that the authors collectively feel deserve more attention. We do, however, discuss the differences between motivation and constraints that consumer VR experience designers have taken on board, compared to traditional academic research.

In Section 2, we give a broader context to 3DUI research and outline

- Anthony Steed is with Department of Computer Science, University College London. E-mail: A.Steed@ucl.ac.uk.
- Tuukka M. Takala is with Aalto University and Waseda University. E-mail: tuukka.takala@aalto.fi.
- Daniel Archer is with Department of Computer Science, University College London. E-mail: D.Archer@ucl.ac.uk.
- Wallace Lages is with School of Visual Arts, Virginia Tech. E-mail: wlages@vt.edu.
- Robert W. Lindeman is with the HIT Lab NZ at the University of Canterbury. E-mail: gogo@hitlabnz.org.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

some specific concerns about 3DUI that are highlighted in the consumer environment. Section 3 then discusses the taxonomy of techniques we present, how the examples were chosen and a description of the goals. The following four sections introduce areas of classic 3DUI concern, Selection (Section 4), Manipulation (Section 5), Locomotion (Section 6) and System Control (Section 7). We add a fifth area of Miscellaneous Techniques (Section 8) that mostly deals with systems or specific situations of the user and their comfort in the system. This is motivated by the framing issues we raise in Section 2. We then conclude and present ongoing activities in this area (Section 9).

2 BACKGROUND AND FRAMING

In this section, we consider how recent work on consumer VR fits within the frameworks and trends of 3DUI. We would emphasise that this is not to criticise either the commercial or academic work, but rather to highlight the slightly different concerns that each has. While in general, design of 3DUI in any context prioritises effectiveness and ease of use, as would any user interface, consumer VR experiences are obviously designed with interesting game play or user engagement as a priority, rather than simply the effectiveness of a technique as a goal unto itself. For example, the design of a game experience places many different constraints on the choice and style of interaction (e.g., visual representation, simple things such as gravity). The first implication of this is that in our later review we have focused on techniques that we think are generalisable rather than working for a very specific game because of the context. The second implication is that techniques work together, but relatively little academic work has looked at the complementarity of techniques, whereas a game experience has to consider the whole design. Thus, we leave consideration of the ways in which design of a coherent experience might affect specific choices of techniques to future work, and here focus on some specific techniques and their differences to what we might call “established practice.”

2.1 Narrow Interpretation of 3DUI

A key observation is that, while there are a variety of consumer VR systems, they form a rather narrow subset of what has previously been considered within the scope of 3DUI [1]. First, the consumer systems that have become popular over the past five years are HMD-based, whereas academic work has focused on a variety of VR systems including immersive projection systems [6], desktop or table systems [7] or novel 6DOF input devices [8]. It might be fair to say that some of that work was done partly because HMDs of the era were not as effective. However, such devices have found niche industrial applications over the past 20 or so years, support different modes of working including working for long periods, and support for multiple users. There is, for example, still a lot of ongoing work on multi-user, but unencumbered displays [9].

A second observation is that consumer systems tend to have very similar configurations for hand controllers. While we will discuss consumer systems without 6DOF tracking for both hands, at the time of writing, there is one dominant form of control for most systems: the user holds two controllers, one in each hand, and these are multipurpose controllers with, usually, a joystick or touchpad, a trigger and several other buttons. Within the broader 3DUI context, there is a lot of work on various forms of hand-controllers with force feedback (e.g., [10]) or controllers that change shape (e.g., [11]). While there are various projects that aim to make new consumer devices such as gloves, advanced controllers, or task-specific controllers, no one device has become very common.

Third, a few aspects of 3DUI are out of scope for this review. While there are options for adding full-body motion capture, these are still very much in the domain of the professional user or serious hobbyist, with relatively few experiences supporting highly articulated avatars. However, we do see this as an emerging area of work. Applications such as *VRChat*¹ suggest that this will become more important going forward, as social VR becomes more prevalent.

¹Here and throughout, if we refer to an application, it will be in bold italics and listed in a section before the references for easy cross-referencing.

The relative narrowness of the 3DUI currently being explored in industry does mean that content developed for one consumer system is relatively easy to port to other systems, requiring perhaps something as simple as a new button mapping. Standards such as OpenXR [12] will possibly allow developers of new devices to back-support older content. However, we expect that there will be a need for new standards that support a broader range of 3DUI for future VR systems.

2.2 Dealing with the Real Environment

It is an over-generalisation, but much 3DUI work done in academia to date has made assumptions about the interface being positioned in a relatively large open space, be it a tracked space in a lab or an immersive projection system. These spaces are probably larger than most consumer VR use cases. Further, consumer systems are often used solo and thus without a helpful lab assistant or student managing cables, moving obstacles and shepherding the user away from collisions. Thus, developers of consumer equipment and experiences have had to plan to deal with the real world as it is found.

This is accommodated, at one level, by the now common use of chaperone systems that are provided by the run-time platform software. These allow the user to delineate their own free space or, in some cases, switch to a fixed position. Experiences thus need to adapt to where the user is within this self-defined space, and keep actions of the user inside the chaperone as much as possible; the user may injure themselves or cause other damage if they stray outside the space without warning. Of course, the free space might change because of temporary objects, or mobile objects (the so-called “cat problem,” where the family cat comes and sits in the free space without being detected by the system or user [13]).

2.3 Accessibility

The issue of broad accessibility is one that has challenged VR content developers over the past few years. Again, it is perhaps reasonable to claim that most academic and prior studies were done with able-bodied users. There has also been a strong bias towards males [14]. One exception is the rehabilitation community, but the focus there has tended to be on the creation of content to facilitate, encourage or monitor exercise (e.g., [15, 16]). The challenge is that the default assumption of many researchers is of an able user, who will stand or sit, and actively move their arms to pick and place objects, etc. Some interfaces require two hands, while others require the user to stand to reach objects, and these immediately exclude users who cannot carry out such actions. Whereas, in the real world, various aids or adaptations might be made, the consumer VR experience is currently quite inflexible.

Accessibility is of course a societal need, so we should expect more emphasis, and eventually standards, on accessibility to VR content. Some direct interventions have been made, such as the Microsoft SeeingVR toolkit that shows how visual aids might be added to applications by the developer or the user [17], or *WalkinVR*, a plugin to SteamVR that facilitates modifications to enable users with limited or uncontrolled motion to use consumer VR systems.

There are broader angles to accessibility, such as dealing with all body shapes and sizes, especially if the environment is suitable for children, supporting people with larger hair arrangements [18], or simply dealing with a desire to sit down when the platform supports this. Environment developers, however, normally anticipate the user will be standing. This issue is linked to the issues in the previous section, because users may prefer to sit to feel safer, or because their environment has more reachable space if the user sits.

2.4 Diegetic Interfaces

3DUIs have excelled at emphasising direct manipulation of interface elements. Indeed the over-arching metaphor of most applications is that the environment affords opportunities to explore, experience situations and then control some aspects of the environment by actions such as manipulation. In many cases, an application might solely be focused on the user interacting with the environment. However, there is often a

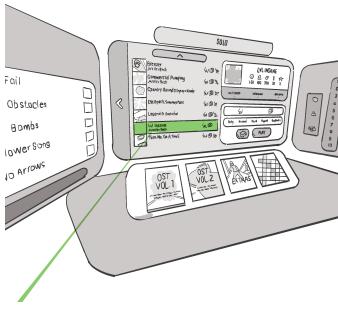


Fig. 1. Multi-screen non-diegetic menu screen in *Beat Saber*



Fig. 2. The projector as diegetic metaphor for a menu in *I Expect You to Die*

need to configure aspects of the experience or call up controls to transition to another location. Such features usually result in the developer constructing some sort of pop-up 2D interface that resembles a dialog screen that one might encounter on a traditional 2D window. These 2D interfaces can pack a lot of controls and are thus quite an efficient use of screen-space. They might include tabs, scrollable windows, sliders, buttons and many other recognisable features. Of course, these features might not be as easy to use in 3D. There are a vast array of options for rendering and composition that are beyond the scope of this article, ranging from flat menus that would not look out of place in a mobile app (e.g., *Beat Saber*, see Figure 1), to 2.5D menus that are mostly flat, but with 3D relief to look like physical controls such as buttons and sliders (e.g., *Rec Room*) through to menus laid out on 3D surfaces that are more effective (e.g., the handheld menu in Google's *TiltBrush*, see also Section 7.3). This latter example also shows the power of using two hands over one for menu interaction tasks [19].

The balance for how much to put into the environment or into a menu overlay is very much up to the designers. Some have pushed as much as possible into the environment, so that the user never need encounter overlays or pop-up windows, whereas others make liberal use of menus. This distinction is sometimes referred to as *diegetic* versus *non-diegetic*, with the use in consumer VR being very similar to the description of the use of the term in first-person shooter games [20]. For example, the game *I Expect You To Die* has multiple levels, so an obvious way to select levels would be to put up a menu where the user selects them. Instead, the game provides a diegetic metaphor where levels that are available appear as film canisters that can be placed into a projector, see Figure 2. In a tongue-in-cheek metaphor, the game *Job Simulator* presents its level selection mechanism as choosing between game-console cartridges in a tray, and having the player insert them into a console to load them. The mechanism to exit the level is to take two bites from the "Exit Burrito," which stretches the diegetic metaphor, but avoids a pop-up menu.

2.5 Physics-based Interaction

The issues of following a diegetic pattern or not is strongly related to the issue of whether the virtual environment should follow rules from the real world. This is not a new problem, and the related issues have shaped 3DUI design from the earliest days. The most obvious issue is that virtual objects cannot have physical properties, and thus cannot exert forces on the user. Virtual objects are thus not "solid" when reacting to touch from the user, be it a virtual tool affording the form of a grip, or a virtual chair supporting the weight of the user. A secondary issue is whether objects obey gravity and other kinematic behaviours, so that while they cannot react to the user completely authentically, they can react with each other.

What is perhaps framed differently for consumer interfaces, is that the repercussions of any lack of physical constraint generates a variety of reliability issues and edge cases. Physics simulations are by their nature difficult to predict and thus random effects might be expected. Typical issues include objects falling out of reach, objects disappearing off to infinity because they escape collision volumes, and glitches with collision (e.g., stuck objects). This makes creating a reliable simulation very difficult; whereas in a lab situation a simulation engineer (or student) might reset a problematic object or level, a consumer experience has to be self-maintaining, or provide for the user to reset things when needed.

We might argue that the solutions to this are generalisable issues, and thus some solutions (e.g., making useful objects that are occluded appear to shine through their occluder as if in x-ray vision, as happens in *I Expect You to Die*) are underscored in later sections, but our feeling is that this is a more general issue of how to determine whether the application is actually in a state where it can progress, and thus is more a systems or programming issue. Indeed, it hints at an interesting theoretical issue of testing about whether a goal is achievable from the current state represented within the game engine.

Finally, we note that consumer applications span the whole range from relying on simulated physical behaviour (e.g., *Gadgeteer*) through to completely fantastic environments that exploit movement, but ignore physics (e.g., *Beat Saber*).

2.6 Likely changes

As a final framing issue, we note that while consumer content has mostly evolved to fit with the dominant types of controller (two 6DOF controllers with a range of buttons), we might expect a variety of changes over the coming years. Generally, new modalities operate as a superset of previous modalities. Thus, while Google Cardboard-style interfaces supported gaze only, or gaze+button interaction, experiences using just these are still playable on later systems, subject to software compatibility.

It seems reasonable that the current state might remain the "entry level" virtual reality experience for a few years. Features such as eye tracking and body tracking can enable new ways of interacting with objects, but they do not need to deprecate older ways of doing things. Hand tracking, though, has a slightly different use, as evidenced by the HoloLens devices, in that it can replace the use of a controller. However, this does not appear to fit with the needs of VR content at the moment. It would be difficult to map all the buttons and joystick input of a controller to hand gestures. Further, a hand controller easily enables multiple actions at the same time (e.g., moving, pointing and grabbing), while the hand gestures to enable different modes might only operate exclusively. Timing and precision of hand gestures is also a poor replacement for the accuracy of a controller, due to the physical nature of the device. Thus, hand tracking raises a range of new issues about the operation of emerging types of consumer content.

3 REVIEW METHODS & TAXONOMY

There are already thousands of consumer applications and demonstrations across the various VR platforms. While some of the main platforms have heavily curated content that focuses on high quality, there are very active independent developers creating either for the more open platforms (e.g., Steam/Windows, or Android), or focusing on hobbyist markets (e.g., Oculus Quest via SideQuest). Each

experience might offer a variety of different user interface techniques, possibly selected by the user. Some experiences may not reveal novel techniques until a substantial amount of experience has been gained, or might only be enabled in certain device configurations. A user might not even notice that a particular interface technique is not something “standard.”

Even in relatively narrow domains such as manipulation, it is practically impossible to systematically review all the options, and thus claim to be able to categorise and analyse them all. As we have shown in Section 2, interaction techniques are also dealing with new issues and thus it remains unclear if older taxonomies can capture the variety of issues to which developers and designers today are responding.

Thus, in doing the background research for this paper, the key idea by the initial team (Steed, Lindeman and Johnsen), was to crowd-source examples of interaction techniques. A Slack channel was created as a working space, along with a Trello board to record ideas. Various social media routes were taken to advertise the crowd-sourcing effort, including advertising to community email lists (3DUI list, IEEEVR list), a number of private Slack channels (including one for VR game developers and another for distributed VR experiments) and social media posts on Twitter. The process was open, with the Trello board serving as a discussion board initially, then a todo list for the writing of this paper, and now as a living resource for readers to engage with.

We had discussions about whether to follow older taxonomies of 3DUIs, what comprised a useful transferable result and whether specific techniques were transferable or not (c.f. discussion of *Exit Burrito* in Section 2). A particular point was whether we should spend a lot of time looking for prior academic art for every technique. In the end, the decision was made that because this was a crowd-sourcing project, we should not try to make a scientific argument about the benefit of a specific technique, and thus while prior art might be found, it would not invalidate the raising of the example as a design exemplar. Thus, while we have made links to related research when appropriate, we have not attempted to systematically compare to prior art. We do encourage readers to add references to related work on Trello (see Section 9).

The Trello board contains many links to video fragments that illustrate the key points in the following sections. Examples that were added after this paper will be clearly labelled. The discussion on Slack is still open and we welcome contributions (see Section 9).

4 SELECTION

In this section we discuss novel implementations or insights into the task of selecting objects. This is a short section, as there are relatively few ways to implement selection in an immersive system, and it is usually combined with techniques for manipulation that are discussed in the next section. It is more important for complex environments where the intentions of the user are hard to interpret, as selection is needed to match a somewhat complex gesture of the user (e.g., pointing) against a scene where it can be difficult to match the target object unless there is some understanding of the role of the user (e.g., see [21]). The “standard” techniques of reaching and touching, or pointing with a ray have seen a lot of adaptation to specific situations. The concept of a bENDY ray was already quite well explored in the 3DUI community [22] and is commonly used in consumer applications. Thus, we consider it out of scope for further discussion and in the spirit of this paper, we describe three examples of techniques suggested from our crowd-sourcing exercise that highlight opportunities for further development.

4.1 Intention-preservation Selection

Selection can be broken down into two separate low-level tasks: Indication and Confirmation (analogous to Pointing and Clicking). Indication is typically done using ray casting or direct touching, while confirmation is often done by pulling a trigger, squeezing the grip or making a fist. When indicating an object out of arm’s reach, a simple technique is to project a selection object (such as a ray, cylinder or capsule) from the user’s hand out into the environment. If this selection object is visible, then indication is relatively straightforward for the user, as there is direct feedback about which object will be selected. However, it might not be so obvious in which direction the selection object will point

depending on the hand shape. This has led to other techniques such as eye-hand ray selection, where the object behind the hand from the user’s view is selected. Questions about which might be preferred have been debated in several papers. We refer the reader to Argelaguet & Andujar’s paper describing the drawbacks of both [23].

In creating the game *Bullet Train* several decisions about how to implement object selection and grabbing had to be made. Donaldson & Whiting discussed this in a talk at the Virtual Reality Developers Conference in 2016 [24]. The game is a fast-paced shooter where the user picks up various weapons. In order to prevent too much bending over, weapons are picked up at short distances (< 2m). Effectively, they are selected first by collision with a capsule attached to the hand, where moving the hand is used to indicate which item will be selected, and then can be grabbed using the trigger. In testing, they found that neither hand-pointing nor hand-eye pointing preserved the intention of all users, in that different users would try to select the items in different ways. Their suggestion was to use the vector between hand-eye and hand-pointing, but to switch to hand-pointing when the user pointed “from the side.” The explicit metric used to switch was not given in the talk, but it would make sense to have a threshold on both the angle of the hand from the eye direction, and the forward direction of the hand from the eye direction.

This is illustrated in Figure 3, where Vector B bisects the forward vector of the hands (vector C in the figure) and the eye-hand vector (vector A). The criteria to revert to hand pointing (Vector C) would be if Vector D (the head direction) and Vector C differ by a threshold.

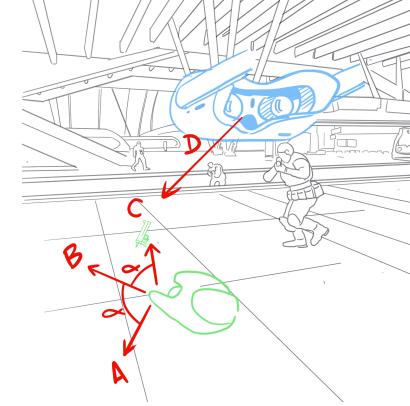


Fig. 3. Selection ray in the *Bullet Train* game. Vector A indicates the eye-hand vector. Vector C indicates the hand forward direction. Vector B indicates the compromise direction that preserves the intention in most cases. Note: In this and future illustrations, blue elements are superimposed representations of the interface devices themselves. Red and solid-black lines are annotations. Green items are the in-environment representations of the controllers.

We note the similarity of this technique to other explorations of how to infer intention with the hand depending on the geometry of the scene. For example, Wagner et al. explore combining hand and ray selection depending on scene structure [25]. The generalisable pattern that might be explored is how the intention of the user can be preserved given knowledge of the environment and the selection actions being performed. A domain where this has been explored before is sketching systems and modelling for VR (e.g., [26, 27]), where the action space is very large, but can be significantly constrained by the local context. What *Bullet Train* suggests is that, even in simple applications or for simple actions, intention might be a more exploitable notion.

4.2 Combined Head and Hand Selection

Another way of improving the usability of selection techniques in VR is to provide multiple ways to control them. In the game *Rez Infinite*, the developers provide the option for a selection method that combines head movement and controller input. The core game mechanic consists of selecting and destroying targets while the camera moves at constant

speed through the levels. In the PlayStation VR version of the game, the controller input is combined with head orientation: the crosshair can be controlled by turning the head or using the thumbstick on the controller, see Figure 4. The game also works with the PlayStation Move controllers, or other 6DOF controllers on other platforms, by using the yaw and pitch angles. Combining hand and head input has a series of advantages. Besides allowing users to choose the method they are most comfortable with, they can also use both of them simultaneously or in sequence. Head pointing makes it easier to select a single object on screen, while the controller allows the user to make quick multi-selections when objects are lined up. It is also possible to use the head to drag the cursor closer to the object of interest and then refine the pointing by using controller input. There are also ergonomic considerations; although pointing with the head is more intuitive, it can be physically demanding. The controller, on the other hand, can stay in a comfortable position during the game, since only yaw and pitch rotations are used to control the crosshair. The technique used by *Rez Infinite* has some limitations. Since the head controls the cursor, it is not possible to look at other objects of interest without drifting away from the current selection group. In addition, since head and controller move independently, the cursor cannot keep a fixed relative position to either one. Sometimes pointing the controller forward will select objects in the centre of the FOV, and sometimes not. This limits the advantage of the natural pointing capability of tracked controllers.

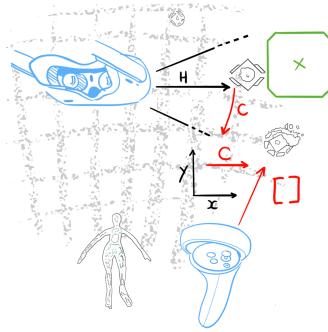


Fig. 4. Combined head and hand selection. The selection crosshair position (Green) is defined by the head direction H and modified by a vector C on the image plane. This vector can be built from either a thumbstick or using Yaw and Pitch angles from a tracked controller. It is also clamped to stay within the headset FOV.

The general consideration here is that in many more abstract applications, selection is the primary goal and thus there is no need to delegate it just to gaze or hand, but to use both if it is efficient. This technique is highly amenable to testbed-style performance evaluation.

4.3 Gaze-dwelling Selection

In some games, in particular those designed for smartphone-based VR, selection is often accomplished by using a gaze-dwelling technique. Gaze dwelling is a well-known technique that uses the head orientation or eye gaze to define a pointing direction [28–31]. Objects are then indicated using ray-casting or another intersection test. To avoid the “Midas Touch” problem [32], targets are only confirmed after a small fixation time, which is used to determine intent. The elapsed time is often communicated to the user through a circular icon that fills or another visual representation, see Figure 5. One example is the game *Wendy*. In this VR witchcrafting puzzle game, gaze dwelling is used to select and interact with items in a room. Gaze selection can also be combined with travel. In *Manifest 99*, gaze is used to pick up objects, and to select creatures and characters for the player to inhabit. In this game, character selection is indicated by a white shader that progressively fills the surface of the object being selected. Besides being a simple and intuitive technique, gaze dwelling creates a distinct experience from using buttons. The game *Land’s End* was initially released for Samsung Gear VR which lacked controllers. However,

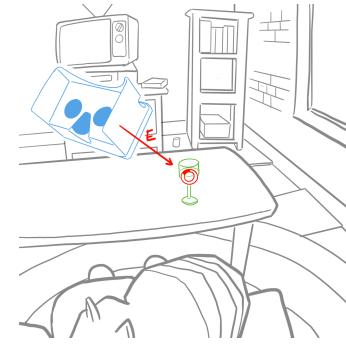


Fig. 5. Gaze-dwelling selection. Head orientation is used to determine a vector for indication (E). The object is selected after the dwelling time has transpired. A circular bar or other visual method is used to indicate how much time is left before the object is selected.

when it was released for Oculus Go (which has a 3DOF controller), the designers decided to stay with gaze dwelling to maintain the experience of moving and solving puzzles only with the eyes [33].

The observation here is that eye-gaze dwell is indicative of user interest and could be a viable selection mechanism in many situations. It also has potential use in future accessible interfaces. The main limitation of this technique is the time required to complete the selection, which limits its use to slow or infrequent selections [34].

5 MANIPULATION

Somewhat similarly to selection, there are not so many options for manipulation with the common configuration of two hand controllers. The obvious approach is to attach the object to the hand [3]. However, as we discuss in the following three sub-sections, there are some interesting new variants that have emerged that deserve further study and use.

5.1 Tomato Presence

The term *tomato presence* was coined by Alex Schwartz of Owlchemy Labs [35] as a description of the grabbing mechanism in their game *Job Simulator*. The concept is very simple: when the user grabs an object, their hand disappears, and they only see the object they are manipulating (see Figure 6).



Fig. 6. Tomato Presence in action. As the user touches the sugar shaker (left), it changes colour. As they grab it (middle), their hand disappears. They can then manipulate the object (right). The hand re-appears when the object is released.

This game involves a lot of object manipulation in fairly close quarters. Tomato presence avoids a common problem, which is how to show the hand of the user grabbing an object: either the object is shown somehow just stuck to the hand, or the object has to be moved into a position where the hand model can be shown to grasp the object. The latter is commonly done in games, but it does mean that the grasp shapes have to be created for many objects and grab configurations.

The technique might be considered controversial because of the prior work that shows that users feel embodiment in avatars with human-like hands [36] and that embodiment can be lower when the user sees an arrow not a hand [37]. However, the consequences of moving the hand are still visible as the object moves. Owlchemy Labs have claimed that few users notice that their hand disappears [35].

Tomato presence is potentially very widely applicable in the most common scenario where the user is represented only by their hands. It would obviously be more difficult to justify if the whole body was being represented. We would highlight the connection to recent work about how avatars can help or hinder interaction. For example, some work has shown that making the avatar partly transparent might aid in typing [38] or make more “avatar friendly” interaction techniques [39].

5.2 Snap Drop Zone

One problem that can arise when manipulating objects is what to do when they are released. Two obvious solutions are just to let them fall to the ground (i.e., obey physics) or to leave them floating in the air (i.e., ignore physics). While some might claim that the more plausible choice would be the former, others would claim that the latter makes for better game play, especially for situations where the player needs to reacquire a released object quickly.

A middle ground approach is to snap the object back into the location and orientation where it was first picked up. This Snap Drop Zone technique (Figure 7) can be very useful, especially for situations where the user stays stationary for some time or is in a vehicle. But whatever approach is chosen, the main point is to build *predictability* into the experience, so that the behaviour doesn’t break the sense of enjoyment or engagement.

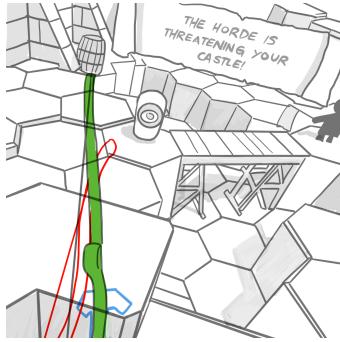


Fig. 7. Snap Drop Zone in the HTC Vive *Longbow Tower Defence* game that comes as part of *The Lab*. The outline of the bow is highlighted when the controller touches it, and releasing the grip button snaps the bow back into place against the wall.

While snapping techniques are somewhat common, especially in construction or engineering applications (e.g., see [40]), Snap Drop Zone is potentially widely reusable because the object can always be found in one or a small number of places if it isn’t carried. This makes it useful for tool-based interfaces. We can also envision this technique being good for things that the user may carry around. For example, if there is a virtual mobile phone, tablet or weapon that might be holstered. Other options are possible, but again, predictability is key.

5.3 Reachability

As discussed in Section 2.2, a particular problem for the consumer VR market is that the application developer has to deal with different sized spaces being available. Many early games were designed to be played from a seated or stationary position because the systems did not have head tracking, or only had limited-range head tracking (e.g., Oculus Rift DK2). For example, in the game *I Expect You to Die*, the scenes are created for stationary experiences, often from a virtual seat. Thus, reachability is not so much an issue, though interestingly that game does have a mechanic for picking up objects that fall out of sight, due to the natural physics used: objects that are needed but are out of sight are highlighted through the object they fall behind. In contrast, some games require larger spaces. For example, the game *Unseen Diplomacy* makes the maximum use of a minimum 4m x 3m tracking space.

For more general environments, bringing everything into reach from a single position is perhaps not possible, but bringing everything within

the chaperone or guardian boundary might be. *Job Simulator* has three different layouts for each of the game scenes depending on the space that is available as reported by the tracking system and chaperone boundary [41]. Thus, the game will work in a space as small as 2m x 1.5m. This required careful thought by the design team to structure and position key interaction work spaces (e.g., counter tops, dish washers) in a relative fashion.

The general observation is that prior work has focused on usability without consideration of reach and interaction with a boundary. Beyond simply using the chaperone, the physical environment could be modelled by scanning [42]. During the writing of this paper, the chaperone system of the Oculus Quest gained an experimental feature to mark up couches [43]. Thus, we should expect more mapping information to be available to the application developer in future toolkits, and it will be interesting to see how this can be exploited by researchers and developers. There are already interesting demonstrations of the possibilities here (e.g., rendering certain real objects into the scene [44] or procedurally generating environments to match tracking spaces [42]).

6 LOCOMOTION

Probably one of the most widely-studied interaction types within the VR academic community is locomotion [45–47]. A locomotion technique attempts to address the problem of mapping movements within a finite physical space to player movement within a potentially infinite virtual space. One way to tackle locomotion is to build a device on which the user walks. This has been the topic of several academic efforts, including robotic tiles [48], and omnidirectional treadmills [49]. Such devices are unlikely to be consumer systems in the short-medium term, but there has been a variety of devices for constrained walking, where the user takes physical strides, but either holds themselves in place or is held in place by a harness (e.g., *ROVR* [50] or *Cyberith Virtualizer*).

For the vast majority of users and demonstrations, the developer will need to provide a locomotion technique based on controller inputs and motion tracking. In this section, we report eight of the techniques reported in the crowd-sourcing effort. We refer the reader to the Trello board (see Section 9) for similar discussions of a further nine (at the time of writing) similar techniques.

6.1 Point-and-Click Walking

The first-person game *VRZ: Torment* contains several locomotion options, one of which is a hybrid between teleport and steering locomotion. The player chooses a destination location with a curved selection arc as is typical in teleport locomotion, but instead of teleporting to that destination, the player viewpoint starts automatically walking towards the destination after the selection is complete. This comes with the downside of the automatic walking being virtual movement, which increases the risk of visually-induced motion sickness. The upside of this form of locomotion is that it is more realistic than teleporting and also requires little effort while the viewpoint is moving towards the destination.

Teleport is now a very common option in consumer VR applications, but there is a worry that it leads to a lower spatial awareness as the user does not have the visual experience of travelling [51, 52]. Thus, the general observation is that techniques such as point-and-click walking might prove to be a happy medium, and might also lead to better spatial understanding by the player than teleportation, because users get a visual transition and can look around during the automatic walking process.

6.2 Narrowing the Field of View

The game *Eagle Flight* is notable for violating an assumption that many have taken on board: that users would get nauseous if moving quickly and turning relative to the horizon. The game has the player flying as an eagle, with visual parallax quite close to the head and with the user performing steep rolling turns. When turning though, the field of view narrows by blurring and a vignette effect around the screen, see Figure 8.

This appears to be a technique that could be applicable in environments with fast or turning motions. Certainly it seems to be an



Fig. 8. Narrowing field of view, as shown in *Eagle Flight*

effect that might be configurable by the user depending on their sensitivity to motion cues. Some academic work and descriptions have recently emerged [53–55]. How broadly this can be used is still an open question.

6.3 Blurred Reorientation

Another technique, blurred reorientation, is similar to field-of-view restriction. Optic flow is reduced via blurring, which subsequently could also reduce illusions of self-motion and motion sickness. In the *Cosmic Wandering VR* demo, the whole screen is blurred during virtual view rotation, while in *VR Tunnelling Pro* and *GingerVR* [56] developer tools offer functionality for blurring only the peripheral vision area.

Display blurring during virtual motion is a technique that can be used to reduce visually-induced motion sickness [57]. One can imagine that if such techniques are proven to be broadly useful, they could become the responsibility of the interface software or hardware (similar in nature to the existence of chaperone systems) in order to facilitate accessibility, rather than being a feature that needs to be handled by individual developers.

6.4 Interaction Scale Adjustment

Changing the scale of the user or the virtual world was explored in academic literature long before the current consumer VR boom [58]. Scaling the user into a virtual giant enables, for example, traversing large virtual worlds via natural locomotion, even in small physical play areas. Conversely, scaling the user to a Lilliputian size gives an interesting new perspective and opens the door for more fine-grained object manipulation in spatial dimensions.

There are many ways to effect a player-scale change. For example, change in scale could be triggered by pressing a button on a hand-held controller, or automatically when the user enters predefined areas [58]. Krekhov et al. examined a locomotion technique where the user could switch between normal and giant mode to affect their stride length in the virtual world [59]. Their user study indicated that in comparison to standard teleport locomotion, their scaling-based locomotion technique enjoyed an increased presence and made the users walk around more frequently without causing cybersickness.

Examples of consumer VR applications with interaction scale adjustment include *Anyland*, *NeosVR*, *Quill* and *The Spy Who Shrunk Me VR*. *Anyland* and *NeosVR* are social VR applications, where individual users can change their size and keep on interacting with others despite size differences. *Quill* is a 3D painting application that allows the user to work at different scales. *The Spy Who Shrunk Me VR* is a stealth game, where the player can shrink themselves to solve puzzles and avoid enemies (Figure 9).

We would make the point that scale changes, while interesting, might have different effects on user experience, such as changes in perception of distance, or even feelings of vertigo. Thus, this needs to be used judiciously, but in many environments there is a need to work on fine detail, as well as to get an overview. See also Sections 7.1 & 8.3.

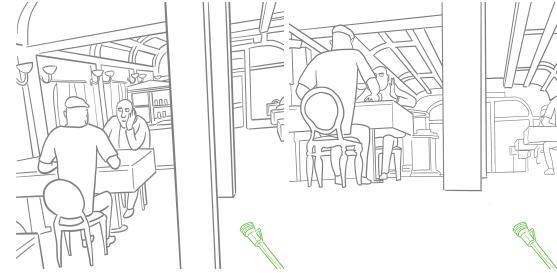


Fig. 9. Scale adjust before and after from *The Spy Who Shrunk Me*, via the shrink-ray tool.

6.5 Rectangular Movement Gain

This locomotion technique was developed for situations in which a virtual object is showcased in the centre of a virtual environment that is larger than the physical play area. It originated in the design for the *Audi AG Virtual Reality Experience*. The idea is to selectively apply movement gain to allow the user to circle around the object of interest and look at it from different angles, even in cases where the object would not properly fit into the physical play area.

The technique works by enclosing the user within the bounds of a rectangle with a predefined size that is smaller than the physical play area and parallel with the horizontal plane. Whenever the user steps beyond this rectangle, movement gain is applied so that the virtual area “slides” in relation to the physical play area.

In Figure 10, the traversable virtual area is marked by boundary B_V and it contains the object of interest O . No movement gain is applied when the user U moves within the rectangle B_R . When the user attempts to move outside the edges of the rectangle with real world velocity V , then the virtual movement is amplified by using a gain constant to multiply the user’s velocity components that are on the floor plane and perpendicular to the edges in question. This results in movement gain V_G and a total virtual velocity \hat{V} . Simultaneously, the rectangle B_R is shifted in the movement-gain direction so that the shifted edges will intersect the (x, z) coordinates of the user’s new position. In other words, the rectangle is dragged with the user whenever they try to step beyond its edges. The boundary B_G in Figure 10 represents the physical play area. If we want to visualize how the physical play area boundary moves with the user in virtual world coordinates, we need to apply the gain velocity V_G to B_G .

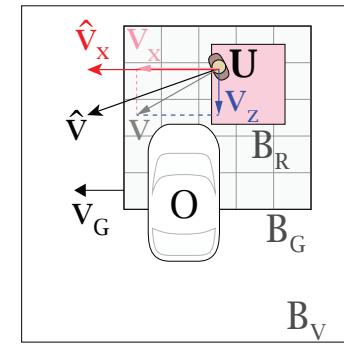


Fig. 10. Example of rectangular movement gain where the user tries to move beyond an edge that is perpendicular to the x -axis. Thus, the gain is applied to the vector component V_x of real-world velocity V , which results in virtual velocity \hat{V} with a vector component \hat{V}_x , so that $V_G = \hat{V} - V$.

This technique is directly useful for any similar situation where there is an object of interest in the centre of the virtual environment. It might be compared to techniques that amplify movement at different scales (e.g., [60]). A general idea for exploration might be scaling techniques that exploit more complex relationships between virtual space and the available tracking space.

6.6 Task-Oriented Teleportation

Teleportation as a general approach has many advantages, as previously detailed. Designers have also come up with variations on the traditional location teleportation, in order to better suit their particular applications. One such variation is to constrain teleportation destinations to specific jump points (Figure 11). This underscores another important point often missed in research studies, which is the important trade-off between precision and fun. While giving players complete control over their movements might seem like a good idea, it could impose unnecessarily stringent precision requirements, leading to frustration and a drop in user enjoyment. Jump points can allow players to focus more on the core game mechanic, and fight less with the control scheme, as evidenced by *The Room VR* experience. An early implementation of a similar scheme was the *Bullet Train* demonstration, where not only was teleportation constrained to specific useful positions, but the rotation of the user was predetermined. This was due to this demonstration being available for the Oculus Rift DK2 which was constrained in its rotational tracking. In *The Room VR*, there is a similar rotation on teleport, but it is more subtle; it is enabled only when the participant chooses a seated mode of operation, given that a seated user cannot turn all the way around.

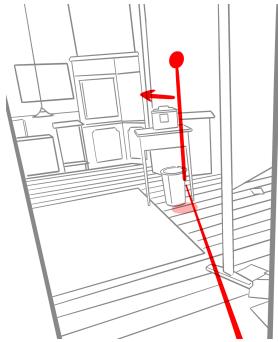


Fig. 11. Task-oriented teleportation, in which a user can only move to fixed points with locked angles in order to perform certain actions, such as *The Room VR*. The red arrow indicates the forward facing direction of the user after the teleport to the point indicated by the stick and teleport ray.

While it appears to diminish user freedom, there is interesting work to be done on whether this has an impact on user preference, situational awareness, etc. It is not difficult to implement within a scene, and it might be possible to analyse scenes to find teleportation locations.

6.7 Rotation Teleportation

Another variation on teleportation includes support for freely indicating the desired orientation of the player once arriving at the designated jump point. This indication of the orientation is normally done with a visual cue, such as the footprints in *Half Life: Alyx* (Figure 12), which rotate with the rotation of the controller about the roll axis. This technique can be slower than the simple Point and Teleport technique, but does provide more control.

This technique solves a few problems, but does require more experience to execute efficiently. It can be used as an extension to any teleportation technique, but certain users might prefer not to use it. It does support seated mode and constrained tracking modes. We would suggest that this is in the realm of options that might generally be offered to users if they are comfortable using it.

6.8 Preview Teleportation

Yet another variation on teleportation, used in the game *Budget Cuts*, allows players to have a preview of the jump point location before actually teleporting. This can give players more information about possible destinations, and might save time if unnecessary teleportations can be avoided.

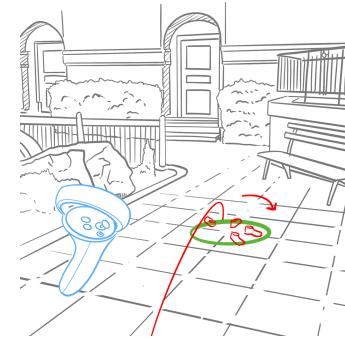


Fig. 12. Rotational teleportation, whereby a user can select the direction they will face by rotating the joystick around the Roll axis during teleportation, as shown in *Half Life: Alyx*

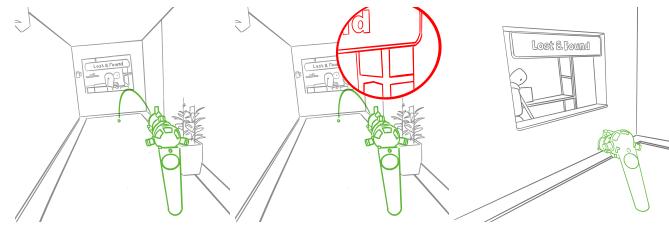


Fig. 13. Translocator mechanic from *Budget Cuts*, giving users a preview window POV from a future location before they choose to move there.

This technique has general utility, and we suggest that it would be interesting to study its general application, especially when it comes to locomotion through complex environments or where locomotion accuracy is important. It might have interesting applications for situation awareness; this is certainly something that the game *Budget Cuts* plays with as part of its design.

7 SYSTEM CONTROL

System control is a broad category covering virtual controls that are embedded within the virtual environment. As noted in Section 2.4, games have sometimes tried to adopt a diegetic approach where system controls are embedded into objects in the world. This is not always possible, especially if there are a large number of controls, or the controls do not have a logical diegetic representation. Thus, many current consumer VR applications include menu or control panels of some sort. Two common strategies are to include panels of controls suspended in space appearing to be in front of the user either at a comfortable distance (e.g., within arm's reach), or a few metres away. The former might be used by touching, the latter by ray selection, see Section 4.

In this section, we discuss three patterns for system control, each of which has been explored in an academic context. It is interesting to see these designs re-emerge, and thus the main question is whether these can be generalised and recommended for certain classes of applications.

7.1 Miniature Worlds

The Worlds in Miniature (WIM) demonstration from Stoakley et al. [61] is a seminal early demonstration of the potential for system control within an immersive environment. The user can interact with the larger environmental context, but also has a miniature copy of part of the environment that they can manipulate. This can bring distant objects within arm's reach and can solve some of the problems of selection and manipulation (c.f. Sections 4 & 5). It is thus no surprise that the technique has been exploited in recent systems [62]. For example, Unity Technologies have demonstrated an immersive scene editor *EditorXR* that has a MiniWorld component for showing parts of the world for editing.

This technique has an obvious place in some applications, especially if they involve a lot of manipulation of object positions. We can, however, make a couple of observations. The first is that unlike most prior work on interaction that dealt primarily with human-scale scenes, a number of demonstrations, for example, *Lucky's Tale* and *Allumette*, use small-scale worlds, where the game or interaction takes place in a relatively small space, with much of the virtual world within arm's reach. This partly solves a problem with interaction at a distance, but also reduces the need for locomotion. Finally, we note that the WIM idea itself can be used as an interesting mechanism within game play, and we refer the interested reader to the game *A Fisherman's Tale*.

7.2 Body-Centred Menus

The idea of using the body of the user as a grounding for placing menus and controls has long been studied in immersive VR [3, 19, 63], and is a design choice that has been made in various games. *Lone Echo* embeds controls into wearable controllers on the user's wrists (Figure 14), fitting nicely with the futuristic theme of this game. It is perhaps interesting to note that a lot designs in this space are based on visions of what augmented reality/heads-up-displays might look like in the future. More grounded in reality is the technique in *RecRoom*, where the main menu is activated by the user gesture of looking at the face of a wristwatch.

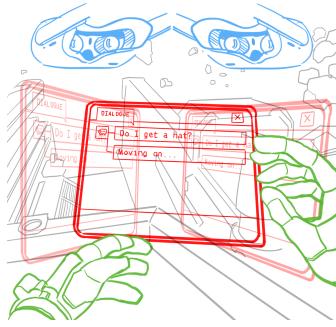


Fig. 14. Body-centred menu from *Lone Echo*.

7.3 In-Hand Menus

As with body-centred menus, in-hand menus are not a novel idea. They build on the natural ability for a user to use both hands to coordinate actions, with the dominant hand performing actions in a coordinate space determined by the non-dominant hand [19, 64]. A body-centred menu also exploits this idea, but an in-hand menu might be constructed as a rather larger tool that can be manipulated by the non-dominant hand, with the dominant hand then selecting items. *TiltBrush* is a good example of this; the in-hand tool is quite large and is manipulated by joystick or buttons on the controller that matches the hand in which it is held.

8 MISCELLANEOUS

The following techniques fall into a class that is not well explored in prior work concerning 3DUI: external factors about the system or its impact on the user (c.f Section 2.2). The techniques below deal with situations that might be outside the realm of interaction techniques themselves, e.g., concerned with user comfort or dealing with the equipment. Thus, they might be more generally used and again we flag that they deserve more study in our field, potentially as part of a broader effort to support practical use and accessibility (c.f. Section 2.3).

8.1 Independent Visual Background

Many VR games include an Independent Visual Background (IVB) for the purpose of reducing visually-induced motion sickness. An IVB provides a visual reference frame that appears to be stationary with relation to the user's physical inertial environment [65].

For example, the game *Windlands 2* features two IVBs that can be enabled by the user: a "comfort cage" that surrounds the player and floor markers. A similar IVB is available in the game *Spire 1*, where the cage is visible only when the player moves virtually. This is also used in the rendition of speed lines in the game *Sprint Vector*. Multiple IVBs are available for Unity and Unreal developers in the free *VR Tunnelling Pro* plugin. Moreover, there are several VR games with cockpits that act as diegetic IVBs, e.g., *Elite Dangerous VR* (Figure 15), *Hover Junkers* and *Vox Machinae*.

Related to IVBs are visual reference frames that are locked to the user's head pose, which could be anything from heads-up displays to virtual noses [66]. An example of the former can be found in the game *Lone Echo*, while *Nintendo Labo's VR Blaster* minigame features a head-mounted gun resembling a nose.

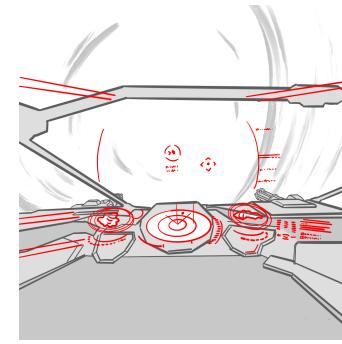


Fig. 15. IVB cockpit from *Elite Dangerous VR*.

8.2 Turn Signals for Cable Untangling

HMDs that are connected via a cable to a PC or a console limit the user's range of movement. If the user keeps rotating several full turns in one direction, the cable can get wrapped around the user's body or chair, or other unwanted tangling might occur. To address this issue, some VR demos such as *Cosmic Wandering* feature visual indicators that show the accumulated user rotation around the Up-axis, which implies cable twisting.

A more general solution is provided by *TurnSignal*, which is a VR utility that can be used on top of any SteamVR compatible application. *TurnSignal* displays a graphical icon on the floor plane that indicates the HMD cable winding direction and magnitude that has accumulated during the lifetime of running the utility. This helps the user to unwind the cable without the need to recall how many full turns they have taken in total.

8.3 Space Adjustment

One of the main issues with accessibility (c.f. Section 2.3) is the height of the player. We have already noted some differences between play for seated and non-seated users. Users might also just be different heights and have different mobility. We have noted several games that provide some means to change scale (Section 6.4). This is not completely generalisable, though, as sometimes the world needs to be experienced at a specific scale. One technique that can be used is simply to allow the user to move the world up and down to compensate for height to make the region that they need to move in more accessible. A good example is *Eleven Table Tennis* where the table can be moved up and down. Another is the game *Brass Tactics*, a game played over a large table top. They combine a locomotion-by-dragging technique, with a technique to move the table up and down by grabbing with both hands.

We can see that such a technique might be useful more generally. For example, sometimes the user wants to switch from sitting or standing, but the system has made an assumption about the reachable area at design time, or estimated user height at start time.

9 CONCLUSION

In this paper, we have described some examples of interesting designs from recent consumer VR games and demos that we feel deserve further study. There are now thousands of consumer experiences available, and to some extent, while there are some good design guidelines and toolkits, best practice is still emerging.

We framed the discussion by reflecting on some of the differences between the demands of consumer VR games, compared to the different foci and stances of 3DUI research over the past three decades. We noted that reliability, fit to space, accessibility and fun all come to the forefront in consumer VR, whereas a lot of academic research has focused on task efficiency. We hope that the designs we highlight prompt new questions and reinvigorate discussion of the main measures and lenses through which we should evaluate interaction techniques for immersive VR.

The paper itself was also partly an experiment in crowd-sourcing. The motivation to look at crowd-sourcing was because of the sheer volume of consumer content to draw upon. Thus, we utilised a process of sourcing suggestions from email lists and other social media. Suggestions were collected on a Trello board that is publicly available at <https://trello.com/b/V1R2xM0u/consumer-vr3dui>. That board serves as an index to other resources, including a Slack channel, and even the source to this paper and its bibliography.

Our primary reflection on the crowd-sourcing process was that engagement levels were very different. A small number of individuals amongst the authors had to push the process along, but suggestions and comments came from many quarters. Engagement was highest where there was a new announcement on some communication channel and there was already a growing list of suggestions. Thus, there may be some bias in that some of the original team seeded the list with their own suggestions. Further, due to the nature of crowd-sourcing, individual contributors will probably highlight what is immediately obvious to them that is similar to what is already listed. Given that many of the contributors are academics, we might expect a slight bias towards work they have themselves contributed to. We don't think that these factors significantly biased the sample, as a broad range of consumer applications is discussed. If we were to run such a process in a similar (e.g., augmented reality) or narrower (e.g., applications of VR in training) domain, the main suggestion we would have would be to seed the data collection with a broad set of examples. Examples of technology domains that we haven't really been able to cover in this paper are the issue of symbolic input, the role of hand-tracking and full-body tracking, and speech and gesture interfaces. The second suggestion is to make it obvious what comprises a contribution (e.g., see the discussion on Trello about what we mean by a "generalisable technique" rather than just an "interesting design").

Finally, one hope we have is that this paper prompts more interaction between researchers and application developers. We encourage readers to add more materials to the Trello board even if it is just a suggestion of an application to look at. There are already many similar examples to the ones we have used in the paper on the board, and there are more examples of applications that use the techniques we have discussed. We hope this list grows as more people engage in this exciting area at the intersection of academia and industry.

PRODUCTS AND DEMONSTRATIONS CITED

All applications, demonstrations, peripherals and videos cited in the document are listed by title rather than author. All were validated June 3rd 2021. Please refer to the Trello board for more examples.

- Allumette by Penrose Studios
<https://store.steampowered.com/app/460850/Allumette/>
- Anyland by Scott Lowe & Philipp Lenssen
<http://anyland.com/>
- Audi AG Virtual Reality Experience by Re'flekt GmbH
https://www.youtube.com/watch?v=At_Zac4Xezw and
<https://www.re-flekt.com/portfolio-item/audi-virtual-reality-experience>

- Beat Saber by Beat Games
<https://beatsaber.com/>
- Brass Tactics by Hidden Path Entertainment
<https://www.hiddenpath.com/game/brass-tactics/>
- Bullet Train by Epic Games
<https://www.unrealengine.com/en-US/bullet-train>
- Cosmic Wandering VR by Punchey
<https://punchey.itch.io/cosmic-wandering>
- Cyberith Virtualizer from Cyberith GmbH
<https://www.cyberith.com/>
- Eagle Flight by Ubisoft
<https://www.ubisoft.com/en-gb/game/eagle-flight/>
- EditorXR from Unity Technologies
<https://github.com/Unity-Technologies/EditorXR>
- Eleven Table Tennis by For Fun Games LLC
<https://elevenvr.com/>
- Elite Dangerous from Frontier Developments plc
<https://www.elitedangerous.com/>
- Espire 1 by Digital Lode
<https://espire1.com/>
- Fisherman's Tale by InnerspaceVR
<https://afishermanstale-game.com/>
- Gadgeteer by Metanaut Labs Inc
<https://gadgeteergame.com/>
- GingerVR by Samuel Ang and John Quarles
<https://github.com/angsamuel/GingerVR>
- I Expect You to Die by Schell Games
<https://iexpectyoutodie.schellgames.com/>
- Hover Junckers by Stress Level Zero, LLC
<http://www.hoverjunkers.com/>
- Job Simulator by Owlchemy Labs
<https://jobsimulatorgame.com/>
- Lands End by ustwo games limited
<https://www.ustwogames.co.uk/games/lands-end>
- Lone Echo by Ready at Dawn Studios
<https://www.oculus.com/lone-echo/>
- Lucky's Tale by Playful
<https://www.oculus.com/experiences/rift/909129545868758>
- Manifest 99 by Project Flight School, LLC
<https://www.manifest99.com/>
- NeosVR by Solirax CoreDev s.r.o.
<https://neos.com/>
- Nintendo Labo's VR Blaster by Nintendo
<https://www.nintendo.com/products/detail/labo-vr-kit-starter-set/>
- Quill by Facebook Technologies LLC
<https://quill.fb.com/>
- Rec Room by Rec Room Inc.
<https://rec.net/>
- ROVR by Wizdish Ltd
<https://rovr.systems/>
- The Lab by Valve
https://store.steampowered.com/app/450390/The_Lab/
- Spring Vector by Survios, Inc
<https://survios.com/sprintvector/>
- The Spy Who Shrunk Me VR by Catland
https://store.steampowered.com/app/754850/The_Spy_Who_Shrank_Me/
- TiltBrush by Google
<https://www.tiltbrush.com/>
- TurnSignal by Benjamin McLean
<https://github.com/benotter/TurnSignal>
- Unseen Diplomacy by Triangular Pixels
<https://unseendiplomacy.com/>
- Vox Machinae by Space Bullet Dynamics Corporation
<http://www.voxmachinae.com/>
- VR Tunnelling Pro by Luke Thompson

[https://github.com/sigtrappgames/
VrTunnellingPro-Unity](https://github.com/sigtrappgames/VrTunnellingPro-Unity)
• VRChat by VRChat
<https://hello.vrchat.com/>
• VRZ: Torment by StormBringer Studios
<https://www.vrzgame.com/>
• WalkinVR by 2MW
<https://www.walkinvrdriver.com/>
• Wendy by Dmitry Kurilchenko and Miles Russo
[https://vrjam.devpost.com/submissions/
36264-wendy](https://vrjam.devpost.com/submissions/36264-wendy)
• Windlands 2 by Psytec Games Ltd
<http://www.windlands.com/>

REFERENCES

[1] Joseph J. LaViola Jr, Ernst Kruijff, Ryan P. McMahan, Doug Bowman, and Ivan P. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley Professional, Boston, 2 edition edition, April 2017.

[2] Mel Slater and Martin Usoh. Body centred interaction in immersive virtual environments. *Artificial life and virtual reality*, 1(1994):125–148, 1994.

[3] Mark R. Mine, Frederick P. Brooks, and Carlo H. Sequin. Moving objects in space: exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 19–26, USA, August 1997. ACM Press/Addison-Wesley Publishing Co.

[4] Robert J.K. Jacob, Audrey Girouard, Leanne M. Hirshfield, Michael S. Horn, Orit Shaer, Erin Treacy Solovey, and Jamie Zigelbaum. Reality-based interaction: a framework for post-WIMP interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 201–210, Florence, Italy, April 2008. Association for Computing Machinery.

[5] Jason Jerald. *The VR Book: Human-Centered Design for Virtual Reality*. Morgan & Claypool Publishers, New York, October 2015.

[6] Carolina Cruz-Neira, Daniel J. Sandin, and Thomas A. DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, pages 135–142, New York, NY, USA, September 1993. ACM.

[7] Maneesh Agrawala, Andrew C. Beers, Ian McDowall, Bernd Fröhlich, Mark Bolas, and Pat Hanrahan. The two-user Responsive Workbench: support for collaboration through individual views of a shared space. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 327–332, USA, August 1997. ACM Press/Addison-Wesley Publishing Co.

[8] Bernd Froehlich, Jan Hochstrate, Verena Skuk, and Anke Huckauf. The GlobeFish and the GlobeMouse: two new six degree of freedom input devices for graphics applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 191–199, New York, NY, USA, April 2006. Association for Computing Machinery.

[9] Juan Sebastian Munoz-Arango, Dirk Reiners, and Carolina Cruz-Neira. Multi-user display systems, Compendium of the State of the Art. *Electronic Imaging*, 2019(3):645–1–645–13, January 2019.

[10] Inrak Choi, Eyal Ofek, Hrvoje Benko, Mike Sinclair, and Christian Holz. CLAW: A Multifunctional Handheld Haptic Controller for Grasping, Touching, and Triggering in Virtual Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 1–13, New York, NY, USA, April 2018. Association for Computing Machinery.

[11] Andrey Krekhov, Katharina Emmerich, Philipp Bergmann, Sebastian Cmentowski, and Jens Krger. Self-Transforming Controllers for Virtual Reality First Person Shooters. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '17, pages 517–529, New York, NY, USA, 2017. ACM.

[12] Khronos Group. OpenXR. <https://www.khronos.org/openxr/>, December 2016. Accessed June 3 2021.

[13] Anthony Steed. VR's "Cat Problem". <https://wp.cs.ucl.ac.uk/anthonysteen/2020/08/vrs-cat-problem/>, August 2020. Accessed June 3 2021.

[14] T. C. Peck, L. E. Sockol, and S. M. Hancock. Mind the Gap: The Underrepresentation of Female Participants and Authors in Virtual Reality Research. *IEEE Transactions on Visualization and Computer Graphics*, 26(5):1945–1954, May 2020. Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[15] Sergei V. Adamovich, Gerard G. Fluet, Eugene Tunik, and Alma S. Merians. Sensorimotor training in virtual reality: a review. *NeuroRehabilitation*, 25(1):29–44, 2009.

[16] Tyler Rose, Chang S. Nam, and Karen B. Chen. Immersion of virtual reality for rehabilitation - Review. *Applied Ergonomics*, 69:153–161, May 2018.

[17] Yuhang Zhao, Edward Cutrell, Christian Holz, Meredith Ringel Morris, Eyal Ofek, and Andrew D. Wilson. SeeingVR: A Set of Tools to Make Virtual Reality More Accessible to People with Low Vision. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 1–14, New York, NY, USA, May 2019. Association for Computing Machinery.

[18] Arwa Michelle Mboya. The Oculus Go Wasn't Designed for Black Hair. <https://debugger.medium.com/the-oculus-go-a-hard-wire-problem-for-black-women-225d9b48d098>, November 2020. Accessed June 3 2021.

[19] Robert W Lindeman, John L Sibert, and James K Hahn. Hand-held windows: towards effective 2d interaction in immersive virtual environments. In *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*, pages 205–212, 1999.

[20] Marcus Andrews. Game UI Discoveries: What Players Want. https://www.gamasutra.com/view/feature/132674/game_u_discoveries_what_players.php, 2010. Accessed June 3 2021.

[21] Anthony Steed. Towards a General Model for Selection in Virtual Environments. In *3D User Interfaces (3DUI'06)*, pages 103–110, Los Alamitos, CA, USA, March 2006. IEEE Computer Society.

[22] Kai Riege, T. Holtkamp, G. Wesche, and B. Frohlich. The Bent Pick Ray: An Extended Pointing Technique for Multi-User Interaction. In *3D User Interfaces (3DUI'06)*, pages 62–65, March 2006.

[23] Ferran Argelaguet and Carlos Andujar. Efficient 3D Pointing Selection in Cluttered Virtual Environments. *IEEE Computer Graphics and Applications*, 29(6):34–43, November 2009. Conference Name: IEEE Computer Graphics and Applications.

[24] Nick Donaldson and Nick Whiting. Going Off the Rails: The Making of 'Bullet Train'. <https://www.gdcvault.com/play/1023648/Going-Off-the-Rails-The>, 2016. Accessed June 3 2021.

[25] Jorge Wagner, Wolfgang Stuerzlinger, and Luciana Nedel. Comparing and Combining Virtual Hand and Virtual Ray Pointer Interactions for Data Manipulation in Immersive Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2513–2523, May 2021. Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[26] Bret Jackson and Daniel F. Keefe. Lift-Off: Using Reference Imagery and Freehand Sketching to Create 3D Models in VR. *IEEE Transactions on Visualization and Computer Graphics*, 22(4):1442–1451, April 2016. Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[27] Andreas Brentzen, Jeppe Revall Frisvad, and Karan Singh. Signifier-Based Immersive and Interactive 3D Modeling. In *25th ACM Symposium on Virtual Reality Software and Technology*, VRST '19, pages 1–5, New York, NY, USA, November 2019. Association for Computing Machinery.

[28] Poika Isokoski, Markus Joos, Oleg Spakov, and Benoît Martin. Gaze controlled games. *Universal Access in the Information Society*, 8(4):323, 2009.

[29] Linda E. Sibert and Robert JK Jacob. Evaluation of eye gaze interaction. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 281–288, 2000.

[30] Vildan Tanrıverdi and Robert JK Jacob. Interacting with eye movements in virtual environments. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 265–272, 2000.

[31] Eduardo Velloso and Marcus Carter. The emergence of eyeplay: A survey of eye interaction in games. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, pages 171–185. ACM, 2016.

[32] Robert JK Jacob. What you look at is what you get: eye movement-based interaction techniques. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 11–18, 1990.

[33] Peter Pashley. Revisiting Lands End. <https://medium.com/@ustwogames/revisiting-lands-end-411e9e44610f>, May 2018. Accessed June 3 2021.

[34] Zhaokang Chen and Bertram E Shi. Using variable dwell time to accelerate gaze-based web browsing with two-step selection. *International Journal of Human-Computer Interaction*, 35(3):240–255, 2019.

[35] Owlchemy Labs. Tomato Presence!

<https://owlchemylabs.com/tomatopresence/>, 2020. Accessed June 3 2021.

[36] Bernhard Spanlang, Jean-Marie Normand, David Borland, Konstantina Kiltzeni, Elias Giannopoulos, Ausi Poms, Mar Gonzlez-Franco, Daniel Perez-Marcos, Jorge Arroyo-Palacios, Xavi Navarro Muncunill, and Mel Slater. How to Build an Embodiment Lab: Achieving Body Representation Illusions in Virtual Reality. *Frontiers in Robotics and AI*, 1, 2014.

[37] Ye Yuan and Anthony Steed. Is the rubber hand illusion induced by immersive virtual reality? In *Virtual Reality Conference (VR), 2010 IEEE*, pages 95–102. IEEE, 2010.

[38] Pascal Knierim, Valentin Schwind, Anna Maria Feit, Florian Nieuwenhuizen, and Niels Henze. Physical Keyboards in Virtual Reality: Analysis of Typing Performance and Effects of Avatar Hands. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI '18*, pages 1–9, New York, NY, USA, April 2018. Association for Computing Machinery.

[39] Diane Dewez, Ludovic Hoyet, Anatole Lcuyer, and Ferran Argelaguet Argelaguet Sanz. Towards Avatar-Friendly 3D Manipulation Techniques: Bridging the Gap Between Sense of Embodiment and Interaction in Virtual Reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, CHI '21*, pages 1–14, New York, NY, USA, May 2021. Association for Computing Machinery.

[40] Daniel P. Mapes and J. Michael Moshell. A Two-Handed Interface for Object Manipulation in Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 4(4):403–416, November 1995.

[41] Owlchemy Labs. Room sizes :: Job Simulator General Discussions. <https://steamcommunity.com/app/448280/discussions/0/37191977174791338/>, 2016. Accessed June 3 2021.

[42] Misha Sra, Sergio Garrido-Jurado, Chris Schmandt, and Pattie Maes. Procedurally generated virtual reality from 3D reconstructed physical space. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology, VRST '16*, pages 191–200, New York, NY, USA, November 2016. Association for Computing Machinery.

[43] David Heaney. You Can Now Mark Your Real Couch On Oculus Quest. <https://uploadvr.com/oculus-quest-couch-guardian/>, 2021. Accessed June 3 2021.

[44] Jeremy Hartmann, Christian Holz, Eyal Ofek, and Andrew D. Wilson. RealityCheck: Blending Virtual Environments with Situated Physical Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12. Association for Computing Machinery, New York, NY, USA, May 2019.

[45] Anthony Steed and Doug A. Bowman. Displays and Interaction for Virtual Travel. In Frank Steinicke, Yon Visell, Jennifer Campos, and Anatole Lcuyer, editors, *Human Walking in Virtual Environments: Perception, Technology, and Applications*, pages 147–175. Springer, New York, NY, 2013.

[46] Costas Boletsis. The New Era of Virtual Reality Locomotion: A Systematic Literature Review of Techniques and a Proposed Typology. *Multi-modal Technologies and Interaction*, 1(4):24, December 2017. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

[47] Massimiliano Di Luca, Hasti Seifi, Simon Egan, and Mar Gonzalez-Franco. Locomotion Vault: the Extra Mile in Analyzing VR Locomotion Techniques. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, CHI '21*, pages 1–10, New York, NY, USA, May 2021. Association for Computing Machinery.

[48] Hiroo Iwata, Hiroaki Yano, Hiroyuki Fukushima, and Haruo Noma. Circulafloor [locomotion interface]. *IEEE Computer Graphics and Applications*, 25(1):64–67, 2005.

[49] Hiroo Iwata. The torus treadmill: Realizing locomotion in VEs. *IEEE Comput. Graph. Appl.*, 19(6):3035, November 1999.

[50] David Swapp, Julian Williams, and Anthony Steed. The implementation of a novel walking interface within an immersive display. In *2010 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 71–74, March 2010.

[51] Andrew MacQuarrie and Anthony Steed. The Effect of Transition Type in Multi-View 360 Media. *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1564–1573, April 2018. Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[52] Kasra Rahimi, Colin Banigan, and Eric D. Ragan. Scene Transitions and Teleportation in Virtual Reality and the Implications for Spatial Awareness and Sickness. *IEEE Transactions on Visualization and Computer Graphics*, 26(6):2273–2287, June 2020. Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[53] Mark Bolas, J. Adam Jones, Ian McDowall, and Evan Suma. Dynamic field of view throttling as a means of improving user experience in head mounted virtual environments, September 2014.

[54] Ajoy S. Fernandes and Steven K. Feiner. Combating VR sickness through subtle dynamic field-of-view modification. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 201–210, March 2016.

[55] Isayas Berhe Adhanom, Nathan Navarro Griffin, Paul MacNeilage, and Eelke Folmer. The Effect of a Foveated Field-of-view Restrictor on VR Sickness. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 645–652, March 2020. ISSN: 2642-5254.

[56] Samuel Ang and John Quarles. Gingervr: An open source repository of cybersickness reduction techniques for unity. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 460–463. IEEE, 2020.

[57] Guang-Yu Nie, Henry Been-Lirn Duh, Yue Liu, and Yongtian Wang. Analysis on mitigation of visually induced motion sickness by applying dynamical blurring on a user's retina. *IEEE transactions on visualization and computer graphics*, 2019.

[58] Regis Koppen, Tao Ni, Doug A Bowman, and Marcio Pinho. Design and evaluation of navigation techniques for multiscale virtual environments. In *IEEE Virtual Reality Conference (VR 2006)*, pages 175–182. IEEE, 2006.

[59] Andrey Krekhov, Sebastian Cmentowski, Katharina Emmerich, Maic Masuch, and Jens Krüger. Gullivr: A walking-oriented technique for navigation in virtual reality games based on virtual body resizing. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play*, pages 243–256, 2018.

[60] Betsy Williams, Gayathri Narasimham, Tim P. McNamara, Thomas H. Carr, John J. Rieser, and Bobby Bodenheimer. Updating orientation in large virtual environments using scaled translational gain. In *Proceedings of the 3rd symposium on Applied perception in graphics and visualization, APGV '06*, pages 21–28, New York, NY, USA, July 2006. Association for Computing Machinery.

[61] Richard Stokley, Matthew J. Conway, and Randy Pausch. Virtual reality on a WIM: interactive worlds in miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '95*, pages 265–272, USA, May 1995. ACM Press/Addison-Wesley Publishing Co.

[62] Kurtis Danyluk, Barrett Ens, Bernhard Jenny, and Wesley Willett. A Design Space Exploration of Worlds in Miniature. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, CHI '21*, pages 1–15, New York, NY, USA, May 2021. Association for Computing Machinery.

[63] Irina Lediaeva and Joseph LaViola. Evaluation of Body-Referenced Graphical Menus in Virtual Environments. In *Proceedings of Graphics Interface 2020, GI 2020*, pages 308 – 316. Canadian Human-Computer Communications Society, 2020. event-place: University of Toronto.

[64] William Buxton and Brad Myers. A study in two-handed input. *ACM SIGCHI Bulletin*, 17(4):321–326, April 1986.

[65] Henry Been-Lirn Duh, Donald E Parker, and Thomas A Furness. An independent visual background reduced balance disturbance evoked by visual scene motion: implication for alleviating simulator sickness. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 85–89. ACM, 2001.

[66] Carolin Wienrich, Christine Katharina Weidner, Celina Schatto, David Obremski, and Johann Habakuk Israel. A virtual nose as a rest-frame—the impact on simulator sickness and game experience. In *2018 10th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games)*, pages 1–8. IEEE, 2018.