An Integrated System for Interaction in Virtual Environments

by

Aubrey Simonson

B.A., Wellesley College (2019)

Submitted to the Program in Media Arts and Sciences
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2021

© Massachusetts Institute of Technology 2021. All rights reserved.

Author . . . .
Program in Media Arts and Sciences
August 20, 2021

Certified by . . . .
Pattie Maes
Alexander W. Dreyfoos Professor of Media Arts and Sciences
Thesis Supervisor

Accepted by . . . .
Tod Machover
Academic Head, Program in Media Arts and Sciences

An Integrated System for Interaction in Virtual Environments

by

Aubrey Simonson

Submitted to the Program in Media Arts and Sciences
on August 20, 2021, in partial fulfillment of the
requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

Many of the systems for interaction currently being used in virtual environments are borrowed from 2D interfaces, such as mobile and desktop computing. These styles of interaction fail to take full advantage of the possibilities offered by immersive environments, and conceptually don't make sense in 3D space. This thesis proposes and evaluates a pair of tools for interacting with virtual environments which are conceptually 3D, draw metaphors from physical reality, and fit together into an integrated system. The first of these tools, Bird, is a proposed solution for selection and manipulation tasks, and the second, Pockets, is a proposed solution for interacting with menus. Additionally, I propose a series of other tools and interaction techniques which follow the goal of designing from a fundamentally 3D rather than 2D position, and which integrate with Pockets and the Bird, but which were not implemented or tested during the course of this thesis.

Thesis Supervisor: Pattie Maes
Title: Alexander W. Dreyfoos Professor of Media Arts and Sciences

Acknowledgments

This work would not have been possible without the support of a large community. They are:

My advisor Pattie Maes, for taking in an additional surprise student and allowing me to stubbornly stick to this work in spite of all of the good reasons not to.

My readers, Misha Sra and Ken Perlin.

The Fluid Interfaces Group as a whole, for being a safe place to land, and Angela Vujic in particular for the mentorship and guidance.

The Object-Based Media Diaspora, and in particular Nina Lutz, who is capable of asking for things and has put out both literal and metaphorical fires during the course of this work, and Dan Novy.

Zach Fredin, for always answering the door to 023 regardless of the hour of night.

Punit Kunjam, for the color picker code.

Everyone who took part in these user studies, and in particular Couch House, who were always willing to test out every prototype.

Jordan Tynes, for introducing me to virtual reality in the first place.

The pika community, and my wonderful housemates.

Dana Gretton, for early hardware help, writing math in LaTeX, being a sounding board more or less every good idea I've ever had, the mid-afternoon pep talks, and dating me for this long.

An Integrated System for Interaction in Virtual Environments

Aubrey Simonson

This thesis has been reviewed and approved by the following committee members

Pattie Maes . . . .
Thesis Advisor
Alexander W. Dreyfoos Professor of Media Arts and Sciences
MIT Media Lab, Fluid Interfaces Group

Ken Perlin . . . .
Thesis Reader
Professor of Computer Science
New York University

Misha Sra . . . .
Thesis Reader
Assistant Professor of Computer Science
University of Santa Barbara

Contents

Chapter 1

Introduction

1.1 Definition of Terms

The history of virtual reality, in terms of duration, is comparable to that of the internet [77]. Over the course of this history, a number of different terms have evolved for similar concepts. For the purpose of clarity, this document defines the following terms:

1. Ray-cast: Any selection technique which allows for control of a ray.
2. Point cursor: Any selection technique which allows for control of a point.
3. Ray cursor: A selection technique which allows for control of a point along a ray-cast.
4. Screen-space: Objects in a virtual environment which are tethered to the camera, and therefore move with the user's head.
5. World-space: Objects in a virtual environment which are tethered to the environment, and therefore do not move with the user.
6. Local working volume/natural working volume: Within arm's reach of the user.
7. User-Assigned: The effect of an input is decided by the user, rather than by the developer/researcher.

8. Inventory: A menu system wherein contents are user-assigned.

9. Simulator Sickness: Also called cyber sickness. The sense of motion sickness which can be caused by the mismatch between vestibular and visual input when interacting with virtual reality.

10. Spatial Computing: As opposed to the definition proposed in Simon Greenwold's 2003 thesis, which required spatial computing to be " meaningfully related to an actual place", I use the term spatial computing in this thesis to refer to computing which interacts with 3D data, wether that data is related to a physical counterpart like GIS data or a photogrammetry scan, or exists purely in virtual space, like the maps of most video games.

1.2 The Desktop Metaphor

In his 1945 paper "As We May Think", Vannevar Bush writes that "The world has arrived at an age of cheap complex devices of great reliability; and something is bound to come of it" [18]. He notes that the solution of cheap, reliable, complex devices exist at the same time as an enormous problem for the scientific community– an overabundance of previous research that was impossible to keep track of by simply reading, organizing, and hoarding text-based documents. He therefore proposes the Memex, a device which includes a number of concepts which would become central to computing decades later, such as hypertext linking. What I find to be interesting and relevant in "Ask We May Think" isn't any of the ideas it introduced which were novel and innovative at the time. It is interesting to me that Bush's information management machine of the future is a desk. Before explaining any of the functions of the device, or how it could revolutionize how we interact with information, Bush tells us that:

> "It consists of a desk, and while it can presumably be operated from a distance, it is primarily the piece of furniture at which he works. On the top are slanting translucent screens, on which material can be projected

> for convenient reading. There is a keyboard, and sets of buttons and levers. Otherwise it looks like an ordinary desk."

David Canfield Smith, who was involved in designing the icons for the Xerox Star, was inspired by the 1970's offices in which he worked. According to [57]:

> "The first thing he did, he said, was to recast his concept of icons in office terms. "I looked around my office and papers, folders, file cabinets, a telephone, and book shelves, and it was an easy translation to icons"".

In many ways, the desktop metaphor was an excellent means by which to interact with something which computers and the introduction of the internet gradually made more and more available: near-infinite information. A desk, for many, was a place to store and interact with information. However, the choice of offices as the source of skeuomorphic design was by no means a neutral one. In "This is Not My Beautiful House", Everest Pipkin points out that:

> "The choice of the office desk as the primary metaphor for this user revolution says a great deal about who these new machines were designed for. After all, those who would directly benefit from the allusion were those already familiar with the office environment. These personal computers were positioned as accessible because they borrowed directly from the logic of day jobs— but certainly not everyone's day job. For instance, there are no commercial interfaces that are modeled after working farms, or assembly lines, or grocers."[59]

They note that this isn't simply a matter of "unmarked users" designing for others who look like themselves out of an ignorance about the larger world about them. The desktop metaphor interacts with certain notions of power relevant at the time of its creation. Pipkin, again, points out that:

> "...it is important to remember that these desktop interfaces were designed in the 1980s when the office desk was a distinct vision of power. It was

> a decade of finance and business, of power-lunches and making deals, of connection and control— a world pointing inwards at the businessman who sat in a skyscraper, making phone calls that had broad consequences."

Whereas computers allowed people the capacity to interact with arbitrarily large quantities of information, virtual reality allows people to interact with arbitrarily large quantities of space. In his 1997 paper, "Moving Objects in Space", Mark Mine notes that the desktop, which assumes a user sitting in a specific location, may not be the best metaphor for interacting with infinite space. A quote from this paper which has guided this work from the outset is:

> "The desktop metaphor further breaks down when the user is inside the user interface. Interface controls and displays must move with the user as he moves through the environment and be made easy to locate. The differences between working in a conventional computer environment and working immersed are analogous to the differences between a craftsman at a workbench and one moving about a worksite wearing a toolbelt. His toolbelt had better be large and filled with powerful tools."[50]

In spatial computing, we have gotten up from the desk, and begun to explore a larger digital world. As the son of a construction worker, this vision of computing which centers around the concept of a worksite rather than the "distinct vision of power" associated with an office desk was inspiring to me. Pockets, the second tool introduced in this thesis (see chapter 3), is inspired directly by tool belts. The tools presented in this thesis center the capabilities of a user who is inside of the user interface, and how they may move around that interface equipped with powerful tools.

1.3 What does a User in VR Need to be Able to Do?

Designing an "integrated system of interaction for virtual reality" is something far beyond the scope of a master's thesis. As such, this document seeks only to explore a few of the most fundamental tasks of interacting with virtual environments. Mark Mine, in 1995, says that the four "fundamental forms of interaction" in a virtual world are movement, selection, manipulation, and scaling [49]. As someone who has never scaled anything in my physical reality, I defined the "fundamental forms of interaction" differently. I considered interacting with objects and moving through space to be the two most essential means of interacting with virtual worlds. Within each of these categories, I defined sub-tasks as follows:

1. Interacting with objects
   (a) Selection/manipulation (specifically rotation and translation)
   (b) Management, organization, and carrying
2. Moving through space
   (a) Moving within a space, such as from one side of a room to another
   (b) Moving between spaces, such as from one virtual environment to another

This thesis proposes, implements, and user tests solutions for each sub-task of interacting with objects. In chapter 4, it will propose unimplemented but potentially interesting solutions for moving through space.

1.4 Design Principles

My vision for the role of virtual reality in society isn't just one wherein virtual reality is a useful, interesting, and usable technology. It's also one in which virtual reality is minimally detrimental to the environment, reasonably accessible for most people, and respects the privacy of users. As such, design choices in this thesis were guided by the following principles:

1.4.1 Modularity

Designs proposed in this thesis are as modular as possible. They should be as easy as possible to only use parts of, modify, and build on top of. This is to allow others who interact with them to be able to modify them, adapt them, and build on top of them. In *Design Justice*, Sasha Costanza-Chock notes that people who are paid to do design usually design for an "unmarked user" – we assume default characteristics about the people whom we design for, and those characteristics are usually based on our own identity [24].

Virtual reality is already inaccessible or unequally inconvenient for certain categories of people, beyond just the cost of a headset. The design of straps for holding HMDs to the head are disproportionately inconvenient for people with long and/or curly hair, who are themselves disproportionately women and people of color. I was a TA for CS321: Mixed and Augmented Reality in the spring semester of 2019, which was taught at Wellesley college. As Wellesley is a womens' college, I got to experience first hand the abundance of long hair caught in velcro and ponytails which had to be taken down or moved to accommodate the headset.

Arwa Mboya, while running user studies focused on Black women in Nairobi, found that the strap design of the Oculus Go she was using was profoundly incompatible with Black women's hair. This inspired her to design a modification to the Go's system of straps which allows it to accommodate a wider set of sizes and shapes of hair [47].

Costanza-Chock also notes that a great deal of design work and innovation is done by "lead users"– people not paid to do design work, but who adapt designs to fit their needs. Via this modularity, I hope to design tools such that lead users and others feel invited to adapt my work to their needs.

Concretely, maintaining this modularity has meant hardware-agnostic design (discussed in 1.4.2), minimizing the number of gestures used, and avoiding assigning specific fingers to specific outcomes. Gesture space, as well as the number of fingers on the human body, is limited. If an action requires a specific gesture, then a user must learn a different gesture for each action to take in virtual space, which does not scale.

1.4.2 Hardware Longevity

All tools proposed in this paper are as hardware-agnostic as possible. I used design schemes with maximally open interfaces to fit the widest possible range of past and future hardware. Rather than requiring a specific, specialized piece of equipment, they are designed to be able to be implemented with a variety of possible hardware elements which users and future researchers are likely to already have. Bird, which is introduced in chapter 2, was implemented using both the optical tracking of the Oculus Quest, and using a dataglove constructed exclusively out of garbage and salvaged parts. Pockets, introduced in chapter 3, can be implemented using any number or arrangement of buttons which are able to communicate to the main application that they have been pressed. This can be new, custom devices, but will also work with a keyboard, mouse, joystick, or any configuration of buttons connected to a microcontroller. This was important partially to avoid motivating the creation of e-waste, and to encourage the re-use of e-waste, and partially to make it more accessible to reimplement these techniques.

1.4.3 Privacy

This work was done with a firm understanding of the role of spatial computing in society, and the potential hazards that come with it. In particular, during the first year of my degree, I was introduced to the concept of "pervasive augmented reality" at a lecture held at MIT by Facebook's Reality Lab Silicon Research Team. By pervasive augmented reality, I mean AR glasses which are designed for continuous, daily use. Augmented reality, by nature of its ability to keep a virtual object still in 3D space while a user moves throughout that space, has to acquire camera data. If this camera data leaves the device, it creates a privacy concern, not just for the user, but for anyone whom the user looks at. Combined with eye tracking, which would be useful as an input method, data collected from augmented reality glasses could create an extraordinarily valuable resource for advertising.

This talk introduced the idea that Facebook's Silicon Research team was interested in making AR glasses lighter and more comfortable to wear for extended periods of time by using the bandwidth offered by 5G to move as much processing off of the devices as possible. Instead, that processing power would be done by centralized servers, owned by Facebook. This turns the glasses into essentially a camera attached to a user's face and worn most of the time, and a screen. While this connection is likely to be extremely secure, it would involve a live video stream of what a user is looking at for most of their waking hours, to be sent extremely securely to Facebook.

In the work presented in this thesis, I've done what I can to avoid contributing to this, and have tried to establish designs which, in the long term, if adopted, I hope will help steer our relationship with technology away from this future. This is discussed more specifically in chapter 3.

1.4.4 Beyond Meatspace

Virtual reality is an inherently cumbersome medium. Even as headsets get smaller and lighter, placing a screen immediately next to your eyes in a manner which blocks out your experience of the world will always be more inconvenient to enter and exit than a quick glance at a less immersive tool, like a phone. It's also hard on the body, causing simulator sickness in many participants, and has the potential to be both physically and emotionally damaging to participants, due to colliding with unseen physical objects or interacting with potentially traumatic virtual content. As such, I don't think that virtual reality should be used in use cases which can be more easily realized without it.

In "Experience on Demand", Jeremy Bailenson encourages developers to not "waste the medium on the mundane"[11]. He suggests that an experience should only be made in virtual reality if it is impossible, dangerous, counterproductive, or expensive to do in physical reality, and that "if five years from now, people are shutting themselves off from the physical world just to read their email, then I have failed in this book."

McVeigh [48] suggests that using virtual reality to exactly recreate physical reality, especially in social settings, is "wasting the medium on the mundane." Rather than investing in increasingly exact recreations of a world which already exists, I'm interested in exploring interactions which are inspired by physical reality, but which go beyond what is possible in the physical world. Designs in this thesis explore reaching for objects beyond the natural working volume, generators which create infinite copies of objects, evocative rather than realistic avatars, and portals through space. These designs have affordances suggested by their forms and our experience of physical reality, but diverge from realistic behavior where realistic behavior becomes limiting. In addition to taking full advantage of the medium via avoiding recreating designs from desktop and mobile computing, this work avoids being overly attached to the specifics of metaphors from physical reality in worlds where physics itself is malleable.

1.5 Chapter Summaries

The remainder of this text is arranged as follows: Chapter 2 introduces related work specific to selection and manipulation techniques in virtual reality, and Bird, a novel tool for selection and manipulation in virtual reality, with 2 prototypes and 2 experiments. Chapter 3 explores related work specific to menus and 3D tools in both traditional desktop computing and virtual environments, and introduces Pockets, a novel means of organizing 3D tools in virtual environments, along with a selection of 3D tools created as part of a pair of user studies. Chapter 4 explores how these tools fit together, as well as presenting a variety of unimplemented related designs. Chapter 5 concludes this work.

Chapter 2

Selection and Manipulation

2.1 Related Work

Virtual environments give users access to infinite space with arbitrary interactive content. Effective manipulation of virtual objects of interest both within the user's natural working volume and at a distance is a principal problem. The two primary categories of egocentric methods for selecting and manipulating objects in virtual environments are virtual hand and virtual pointer metaphors [61]. Virtual hands, or direct manipulation, are any technique which maps the user's real hand to a virtual representation of it somewhere in 3D space. Virtual pointers, or ray-casts, allow the user to interact with the environment by controlling one end of a ray which can be used to point to objects.

Ray-casting techniques allow users to interact with objects at a distance very quickly and precisely [31][61]. However, because they describe a line, rather than a point, ray-casts provide no way for a user to specify which object they want to select without an additional disambiguation step if that line intersects multiple objects [31]. Many techniques have tried to solve this problem, which we henceforth refer to as "the laser pointer dilemma", by using separate disambiguation steps, heuristics, and additional inputs. Ray-casting techniques, because they specify a ray rather than a point, do not allow the user to bring an object closer to or farther away from them along the length of the ray, henceforth referred to as "manipulation along the z-axis".

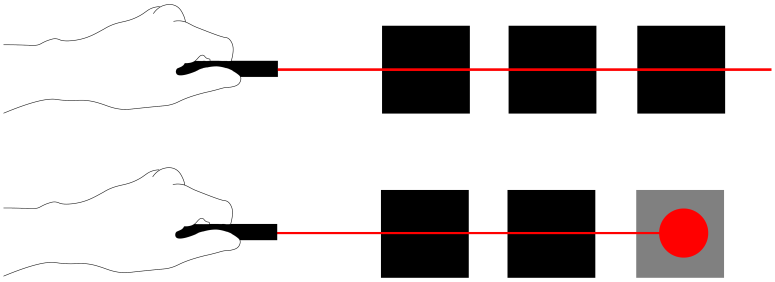


Figure 2-1: Because they describe a line, rather than a point, ray-casts provide no way for a user to specify which object they want to select without an additional disambiguation step if that line intersects multiple objects.

Many techniques, including the Bird, introduced in this thesis, have tried to solve this problem by including a point cursor along the ray.

Ray casting techniques also do not allow for reliable rotation of objects. Rotation along the z-axis (defined as the distance from the origin of the ray) is possible by rotating the origin of the ray itself, for example, by rotating the wrist. However, rotation along the other two axes is always around the source of the ray, rather than around the center of the object, or a point closer to the object.

Direct object manipulation techniques can allow for rotation and translation within the user's natural working volume using only one hand, but cannot reach beyond the natural working volume without some variety of abstraction. Object manipulation based on literal hand position also poses problems in maintaining the immersiveness of the user experience. Existing direct object manipulation techniques set an expectation for haptic feedback when the object representing the user's hand collides with the object being manipulated. When this expectation for haptic feedback is refuted, users feel less present in the virtual world [72]. A large body of research into providing haptic feedback to the hand when touching an object in 3D exists [64] [78], but integration of haptic feedback into direct object manipulation techniques is still immature. There is a need for techniques that abstract from the hand to give users greater reach and avoid haptic inconsistencies.

Finally, most modern interaction with virtual environments is through some va-

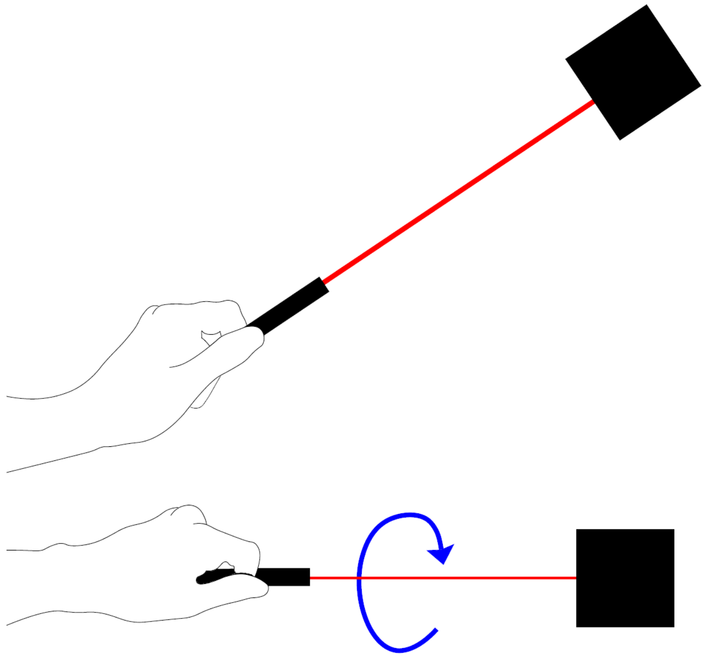


Figure 2-2: Ray-casting techniques allow for a user to manipulate the roll of an object by rotating the wrist which controls the ray. Rotations of the wrist in the yaw and pitch instead cause translations of the object.

riety of handheld controller featuring buttons, and most modern interaction design is therefore done with the assumption of the presence of a controller and buttons. However, the recent release of hand tracking on the Oculus Quest suggests that fully controller-less interactions are becoming standard. Interaction techniques in this controller-less future would need an intuitive way for users to select and interact unambiguously with objects at any range in the absence of buttons.

Several techniques solve the laser pointer dilemma by including a separate, specific disambiguation step in which the user specifies which of the intersected objects to select. SQUAD (Sphere-casting refined by QUAD-menu) involves initially defining a volume via a ray-cast, and then distributes all selectable objects in that volume into four clusters [42]. The user specifies which cluster the desired object is in, and the cluster is then distributed into four smaller clusters, until a cluster which contains only one object is selected. This method can be used to select items in dense environments without requiring the user to be precise with their selection at any point in time, but it also requires several steps of disambiguation. The flower technique, originally designed for volumetric displays, is a ray-cast technique inspired by Ramos 2006's Tumble, Splat technique [62] [31]. If a ray passes through several selectable objects when a button on the device is pressed, it animates them moving into a flower-shaped menu around the ray. The user can then move the ray to the specific object which they would like to select. Users of this technique had some difficulty with following where an object moved in the flower array. The lock ray, also a technique originally designed for volumetric displays, involved selection with a ray-cast [31]. When the correct object was intersected by the ray, the user could push a button to "lock" the ray in place, and then move the controller either forward or backwards to indicate which object along the length of the ray to select. Outline Pursuits is a technique which uses eye tracking for its disambiguation phase [68]. Users make an initial selection with a ray-cast, and if the ray passes through multiple selectable objects, each of these objects is highlighted, with a mark which moves around the outline of the object at a specific rate. The user follows the mark on the border of the object with their eyes in order to select it.

Point cursor techniques, or ray casting techniques which involve controlling a specific point along the ray, solve the laser pointer dilemma because the point should only intersect one object, while the ray may intersect several. They also allow the user to manipulate the selected object along the z axis if the user is able to control the location of the point cursor along the ray. In one study, indirect depth control of the point cursor using buttons was shown to have a faster selection time than other methods [14]. However, the cursor moved at a linear rate along the length of the ray, which could become frustrating to users as they became more skilled with the technique. RayCursor is a technique which combines indirect point control of a point along a ray using buttons with a heuristic technique [12]. Jedi ForceExtension [21] moved objects along the length of the ray at a constant speed, using a combination of button press and head pose, but used only a standard ray-cast for selection. The Depth Ray is a ray cursor designed for use in volumetric displays [31]. The user can control the position of a point cursor along the length of the ray by moving the hand forward or backward, with a 1:1 control - display ratio. While this was a very effective technique in the small space of a volumetric display, it has not been tested in larger virtual immersive environments. iSith specified an individual point in 3D space by using two ray casts. One ray was controlled using each hand, and the point cursor was the point of intersection between the two rays [82]. iSith allowed users precise control in their natural working volume, and the ability to easily move objects which were farther away, but required two devices with 6-DOF tracking, and the use of both hands.

The FlexPointer [28] notably solved the laser pointer dilemma by allowing the user to bend the ray around intercepting objects using an additional input device, rather than controlling a point along the ray or using a separate disambiguation step. This technique notably required two hands to operate.

While direct manipulation, or virtual hand metaphors, make selection unambiguous and both rotation and translation on all axes of an object possible, in its most basic form it does not afford reaching beyond the natural working volume. Mine et. al tried to solve the difficulty with grabbing objects which are far away and bringing them into the user's natural working volume using the "scaled world grab" [50]. The scaled world grab brought objects beyond the natural working volume into the user's hand by shrinking the entire world, and then returning it to its usual scale when the user released the object. This tended to cause a higher rate of symptoms of simulator sickness than other techniques [14]. Go-go is a technique which maps the hand non-linearly, such that a partially extended arm keeps the virtual hand and real hand in roughly the same location, but a more fully extended arm sends the hand much farther away [60]. This technique allows users to interact with a larger sphere than their own arm span, while also retaining the control of direct object manipulation and without introducing any explicit mode switching. However, it leads to shoulder and arm fatigue due to its reliance on whole-arm motion. The Elastic Arm is a more recent adaptation of the Go-go technique [3]. When the arm is moved beyond a certain distance away from the body, the virtual hand moves away from the body in a rate-controlled way, rather than using a non-linear mapping.

Attempting to combine the selection speed of ray cursors and the intuitive control of direct object manipulation, Bowman et. al designed the Hand-centered Object Manipulation Extending Ray-casting, or HOMER technique [13]. Via ray-casting, the user placed the cursor representing their virtual hand near the intended object, and then directly manipulated that object at a normal scale, having effectively moved their local working volume to the location of the object. When they let go of the object, the hand returns to the same location as its real-world counterpart. While this technique allows for direct object manipulation at a distance, because it incorporates a ray cursor, it still could not disambiguate between different objects intersected by the ray. It is also not a particularly useful tool for grabbing objects which are far from the user and efficiently pulling them closer, because a user's movements remained on a one-to-one mapped scale, even when their hand was far away.

Overall, the body of existing abstracted object manipulation techniques is characterized by a proliferation of extra interaction mechanics, all aimed at resolving a persistent conflict between disambiguation, use beyond the natural working volume, and ease of use.

2.2 Bird

2.2.1 Concept

The Bird is a point cursor technique which allows the user to control the position of a cursor along a ray with one hand. The cursor is attached to the hand, which allows the user to manipulate it directly. The user can control the distance between the point and the hand by curling and extending the fingers. When the fingers are fully curled, the cursor should be at roughly the same position as the hand, and when the fingers are fully extended, the cursor is sent far out in front of the hand. As with a 2D mouse, one finger is reserved to click, and is not included in calculations of the position of the cursor. This is where the name “Bird” comes from– it's a mouse that flies.

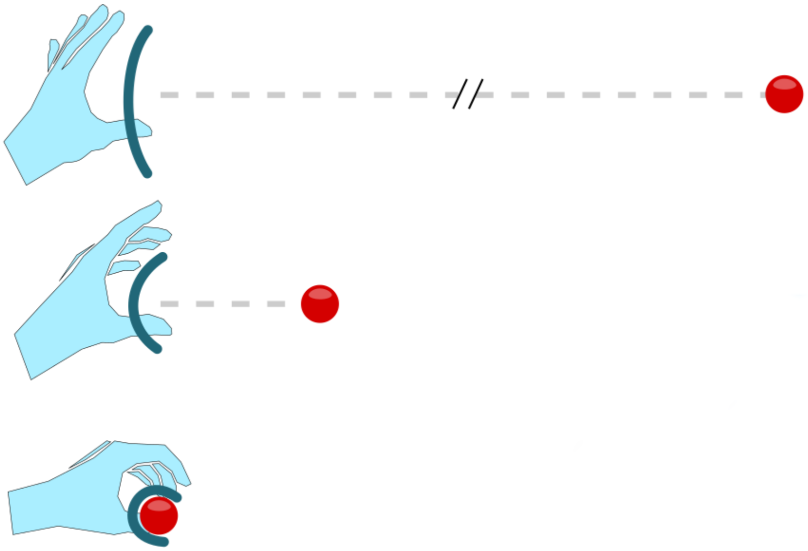


Figure 2-3: Opening the hand sends the Bird cursor farther away, while closing it brings the cursor closer.

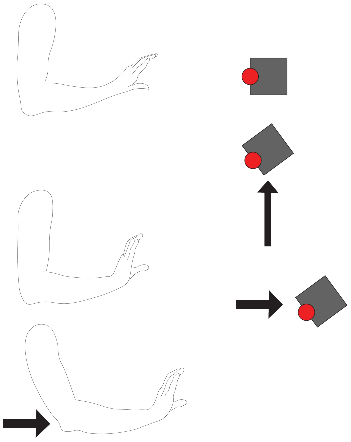


Figure 2-4: Small rotations of the hand using Bird can be used to cause large translations of the held object. Small translations of the held object can be made by moving the arm.

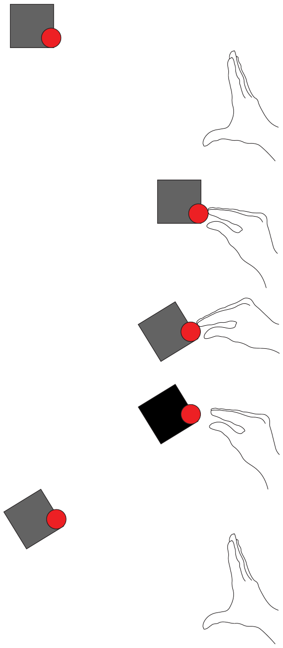


Figure 2-5: The bird can be used to rotate objects in the pitch and yaw more easily than other point cursor techniques, because bringing the held object close to the source of the ray (and therefore center of rotation for both of those axis) is simple.

The Bird was inspired by the actual gesture I found myself making when working on a different virtual reality project and trying to describe how I wanted to be able to interact with objects at a distance. While explaining that I wanted to be able to simply reach out and grab things, I noticed myself making a grabbing gesture, and realized that making a grabbing motion with one hand at an object could allow me to do that.

While the Bird, as a ray cursor method, cannot be used for rotation around the pivot of the held object, it can be used to very quickly bring objects close to the hand. This more closely matches the location of the center of rotation of the hand to the location of the center of rotation of the object. Combined with a clutching motion, it can be used to rotate objects in pitch and yaw. This clutching motion for rotation is depicted in Figure 2-5.

Mine et. al noted that the absence of expected haptic feedback when touching an object also makes directly manipulating objects difficult for the user[50]. Because, as a

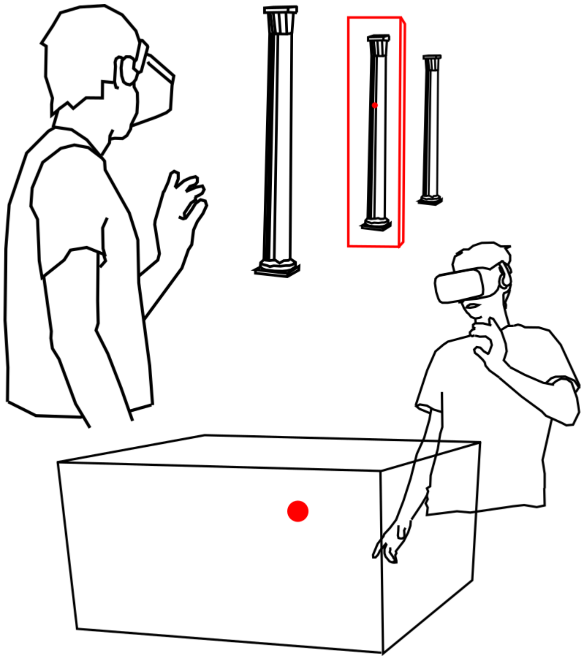


Figure 2-6: The Bird can be used to control a point in 3D space, and that point can be used to select objects.

metaphor for interaction, the Bird is separate from the physical hand, it provides high agency, or sense of responsiveness to user will, without enforcing ownership, or sense that the Bird is the participant's physical hand. High agency has been shown to be associated with high performance in direct object manipulation tasks with abstract cursors, while ownership intensifies negative effects of missing haptic feedback [5]. Intuitively, a traditional 2D mouse has a similar relationship to the user's hand, in that it is not usually disruptive for the mouse to run into and stop at the edge of a screen despite not "feeling" the collision. The Bird also avoids the disorientation and disruption of virtual presence caused by the absence of haptic feedback when an object is touched [69][6] by having the user touch objects with an abstract cursor, rather than one which they identify with as their hand.

2.2.2 Implementation

The Bird can be implemented in a variety of ways, and has already been prototyped using both finger tracking via a custom-made data glove, and using camera-based hand tracking. The Bird requires only a few degrees of freedom of the hand to be measured precisely. It only requires a number to represent the overall "openness" of the fingers, which ideally takes partial influence from all fingers (other than the one being used for clicking); accurate data about the 6-DOF position and rotation of the user's palm; and an indication of whether or not the user's clicking finger is sufficiently bent to indicate a click.

2.2.3 Prototype 1

I initially chose to implement the Bird using a custom data glove, rather than existing depth-cameras, because current depth-camera technology was still too imprecise to detect fine motor motion (in the case of the Microsoft Kinect and Intel Realsense), had too small of a view area (in the case of the Leap Motion), or was too unstable to prevent user frustration due to loss of tracking and/or incorrect tracking (in the case of all three). Fortunately, any lightweight electromechanical device lying on the back of the fingers or hand that takes one high-precision, low-accuracy extension measurement per finger is sufficient to implement Bird. As I show, this kind of interface can be very inexpensive and accessible. I chose this route for the purpose of our initial study, but no such device is strictly required, as I demonstrate with prototype 2.

My initial prototype was implemented using a custom, lightweight data glove using ten 2.2" Spectra Symbol FlexSensors from SparkFun, an Adafruit Feather, disposable packaging from Hamilton pipette tips, and a winter glove. The FlexSensors are secured to the back of the user's fingers via sheathes of disposable packaging sewn to the glove, which causes the sensors to bend with the fingers, allowing us to read a voltage difference between when the fingers are fully extended, and when they are bent. Only a single unitless number representing the degree of bend for each finger is required, obviating the need for expensive, noisy, or range-limited finger tracking. Real time bend measurements were communicated to the computer via USB cable. The position and rotation of the palm was tracked by attaching an Oculus Touch controller to the back of the hand, slipping the knuckle guard around the fingers.

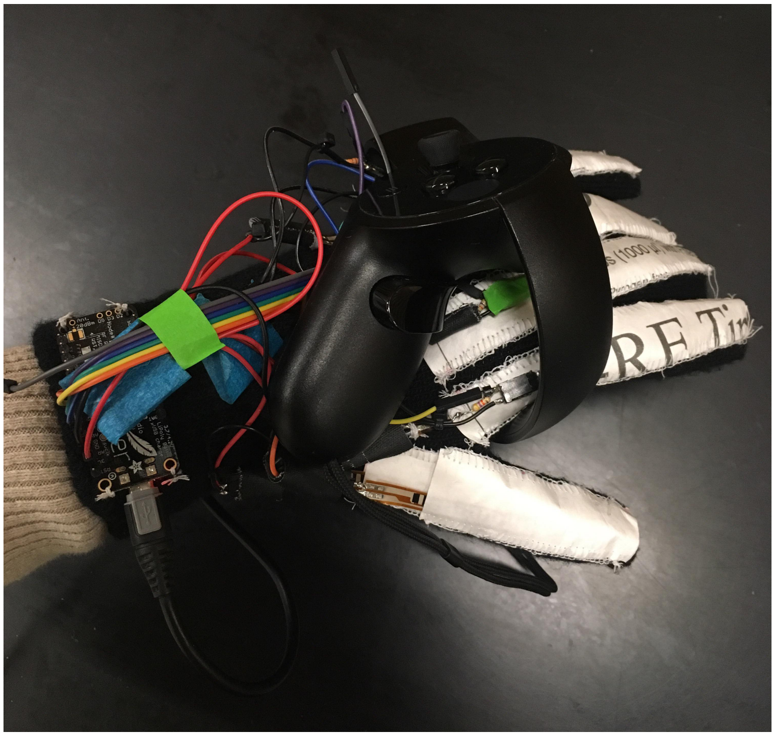


Figure 2-7: The hardware involved in our first implementation of the bird, which was not used in our user study.

Any lightweight electromechanical device lying on the back of the fingers or hand that takes one high-precision, low-accuracy extension measurement per finger is sufficient to implement Bird.

I based the position of the cursor on the curl of the participant's fingers. I established a maximum range which the user should be able to reach, and continuously recorded the largest and smallest voltages which we read off of the glove in order to dynamically calibrate the range of voltages expected from each finger. I also eroded this range by a small amount each frame, to prevent the tendency for the range to expand over time as the user makes occasional acute movements which are hard to repeat intentionally. The position of the bird was therefore the palm position extended out from the palm by a distance $d(t)$, calculated according to

$$\mathrm{V}_{i,max}(t) = \max_{t' \leq t}(V_i(t), V_i(t') - \epsilon \Delta t_{i,max})$$

$$\mathrm{V}_{i,min}(t) = \min_{t' \leq t}(V_i(t), V_i(t') + \epsilon \Delta t_{i,min})$$

$$\mathrm{b}_i(t) = \frac{V_i(t) - V_{i,min}(t)}{V_{i,max}(t) - V_{i,min}(t)}$$

d(t) = $C\sum_{i=2...5} b_i(t)$

where $V_i(t)$ is the measured voltage for finger i at time t in arbitrary units, ϵ is the small coefficient that governs the rate of range erosion, and $\Delta t_{i,max}$ and $\Delta t_{i,min}$ are the times since the minimum ($V_{i,min}(t)$) or maximum ($V_{i,max}(t)$) for the range of finger i were updated by an extreme small or large measurement, respectively. $b_i(t)$ are the normalized (unitless) bend values. The result of the calculation, the cursor distance from the hand at time t, represented by $d(t)$, is the sum of the bend values $b_2, ...b_5$, which includes all fingers except the pointer finger, scaled by an arbitrary fitting constant C to reach the desired maximum range.

I implemented clicking by measuring the voltage across the bend sensors attached to the pointer finger, and comparing it to a "click threshold". A click occurred any time the extent to which the pointer finger was curled, as measured by the sensor voltage, was greater than the click threshold, after having been less than the click threshold. Because holding the pointer finger curled continuously in order to hold onto an object was uncomfortable, I instead had participants click once to select whichever object they had last touched, and click again to deselect it. I called this selection technique a "sticky click". Objects which were selected were held in the same manner as they were for the other three techniques. Whichever object the cursor had last touched, which was therefore the one which the user would select were they to click, was indicated with the same highlight as was used in the other three techniques.

2.2.4 Experiment 1

Task I compared Bird to three similar techniques found in the literature by having participants complete a sorting task in a virtual environment with each of them. The sorting task featured 30 randomly generated objects scattered throughout a space larger than the natural working volume, in a variety of colors and shapes, and with a variety of words on them. Participants were asked to sort objects into between 2 and 5 groups using any attribute of them which felt salient. They were told that the sorting task itself was designed to give them a structured reason to grab objects which were very far away, bring them into their natural working volume,

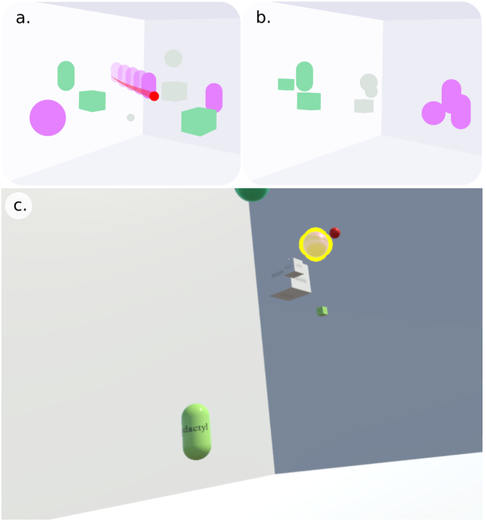


Figure 2-8: Sorting task. (a) Participants were given a random collection of 30 virtual geometric objects of various shapes, sizes, colors, and text labels. Using the cursor (red) controlled by each technique in turn, they were asked to sort the objects into 2-5 groups by any criterion (b). Each user repeated the task 4 times, once with each technique, in randomized order. (c) Screenshot from the perspective of a participant. The user-selected object is highlighted with a yellow border.

and move them to arbitrary locations beyond the natural working volume again. We asked participants to use their left hand (non-dominant for most participants) for all interaction techniques, in order to exacerbate fatigue, which otherwise might not be as perceptible in such a short task. The order in which users were presented with different techniques was randomized to account for learning effects, and users were presented with a brief survey about the technique after each trial.

Participants and Apparatus 15 participants (age range 22-30, 8 women, 7 men, 12 right- handed, 2 left-handed, 1 ambidextrous) were recruited. 13 had experience using a VR device, and 2 described themselves as very experienced with virtual reality. Participants received no remuneration. Participants interacted with the task using an Oculus Rift Head Mounted Display. The experiment application was written in Unity 2018. Participants used an Oculus Touch controller for all techniques.

Other Techniques Bird is a technique for direct object manipulation, which allows users to reach beyond their natural working volume, move objects from beyond their natural working to within their natural working volume, and from within their natural working volume to beyond their natural working volume, with only one hand, and with no snapping to specific points in the environment. I conducted a review of the literature to find other techniques which also have all of these properties, and found three. These three techniques are the Go-Go technique, the Scaled World-Grab, and an extension of ray casting which allows the user to specify a point along the ray using button input [28][50][14]. I considered HOMER as an additional fourth technique, but decided against it due to the fact that it isn't optimized for moving an object from very far away to very nearby, or vice-versa [21]. Moving an object a large distance with HOMER would involve using a ray-cast to specify the object, moving it a distance linearly mapped to the user's hand, and then re-selecting it with the ray-cast in order to move it again, in a clutching motion similar to using a low sensitivity traditional mouse to traverse a very large screen.

Each of the three historic techniques were recreated for the purpose of this study. Because all three of these techniques were designed between 1996 and 2001 for much older platforms than the Oculus Rift, I could not use exactly the original code. Instead, each was implemented exactly as they were defined in the papers published about them.

1. I defined GoGo as a technique which uses "a non-linear mapping between the controlled motion of the user's hand in the real world and the effected motion of the virtual hand in the immersive environment" to create "seamless direct manipulation of both close objects and those at a distance" [50]. I implemented this technique by measuring the distance between the user's controller and headset. The cursor was visualized as a roughly fist-sized red sphere. The position of the user's cursor was linearly mapped to the position of their controller while their hand remained within "close range", and when their hand moved beyond this perimeter, their hand moved farther away from their body exponentially. When the cursor was positioned within an object, that object was highlighted with a yellow tint and outline, and the object could be "held" with the cursor by holding down the trigger button on the controller. When the object was held,

it maintained its position relative to the cursor, it became rigidly bound to the cursor by both position and rotation, while the cursor could still be moved.

2. I defined the Reel technique as a ray-casting technique which "moves the hand to the object", and which uses "buttons to move the object closer or farther away" [14]. In my implementation, the Y button of the controller could be used to move the cursor farther away from the hand. The X button of the controller could be used to move the cursor back towards the user. All visualizations and mechanics for holding an object worked in exactly the same manner as in the Go-Go technique.

3. I defined the Scaled World Grab as a tool by which "selected objects that fall outside of the reach of the user are brought instantly into reach by automatically scaling down the world about the user" [50]. To simplify the implementation of this technique, we scaled the user in the scene relative to the distance between their controller and headset. All visualizations and mechanics for holding an object worked in exactly the same manner as in the Go-Go and Reel techniques.

For all four (three historical, plus the Bird) techniques, participants heard one clicking noise when an object was picked up, and a different clicking noise when the object was put down.

Results

Enjoyment Aside from the Scaled World Grab, which participants described as *"hard"* and *"embarrassingly difficult,"* enjoyment across different techniques was relatively flat (Figure 2-9). On a 5 point Likert scale, Bird was about as enjoyable to use as other techniques from the literature.

Frustration Participants rated their overall frustration with each technique on a 5 point Likert scale (Figure 2-10). While the Bird was a notably frustrating technique, I believe that much of the frustration which users experienced with it may have been related to limitations of the custom data glove, which is discussed in a later section.

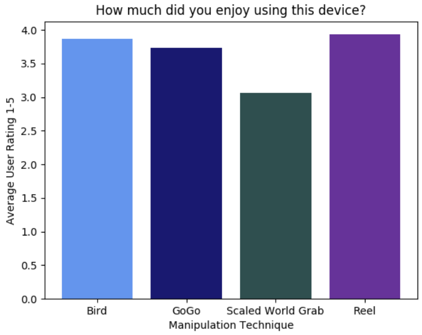


Figure 2-9: Average user response to the question "How much did you enjoy using this device?" on a scale of 1-5.

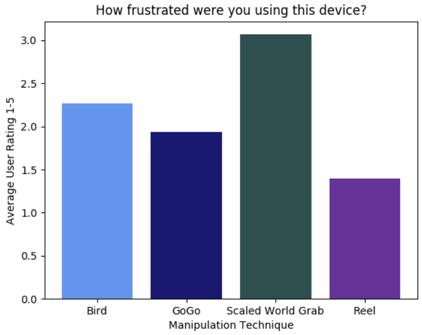


Figure 2-10: Average user response to the question "How frustrated were you using this device?" on a scale of 1-5.

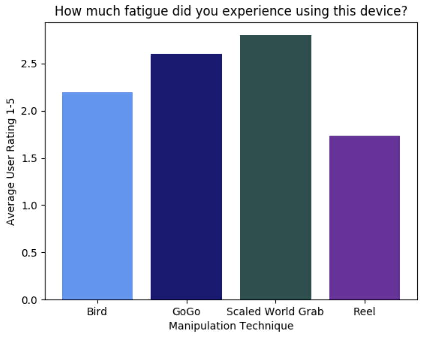


Figure 2-11: Average user response to the question "How much fatigue did you experience using this device?" on a scale of 1-5.

Fatigue On average, users rated Bird better than all techniques other than the Reel in fatigue (Figure 2-11). Compared to other techniques, Bird was fatiguing in different ways. The Bird tended to tire the wrists and fingers, while Go-Go tended to be much more fatiguing for the full arm. One user described the Go-Go technique as *“a workout.”* Another noted that, when using the Bird, *“For far things, I can reach them but my hand feels a little sore if I have to do it a bunch of times.”* This may be mitigated in future implementations by shifting the way that the Bird’s position was mapped onto hand openness, so that the maximum distance is achieved without straining the fingers.

In the current implementation, the hand was usually held very open, and it was never necessary for the user to make a full fist. If a mostly open hand were to map to a far reach, and the cursor was made to actually correspond with the hand with a closed fist, this type of fatigue may be reduced.

Presence Immersion metrics were calculated using the three questions related to presence from Bob G. Witmer’s 1998 paper “Measuring Presence in Virtual Environments: A Presence Questionnaire”[81]. Users were asked to rate each of these questions on a 5 point Likert scale, and I took the average of their answers to measure their overall immersion (Figure 2-12). Scaled World Grab seemed to break immersion. All other techniques rated at a medium level. Generally low immersion may be attributable to what one participant referred to as a lack of *“’placeness’ in any of the*

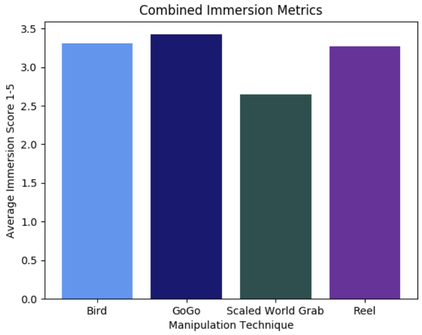


Figure 2-12: Average user response to three questions regarding presence. Each of these three questions were on a 5 point scale.

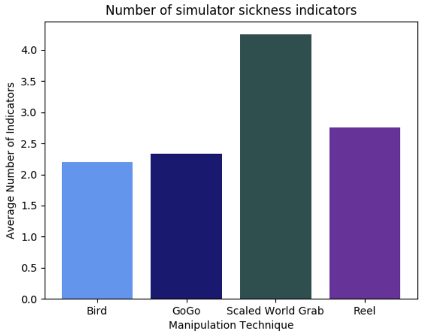


Figure 2-13: Average number of symptoms of simulator sickness (out of 16) experienced by the user after each technique

[tests], mainly because the environment is an abstract white box."

Simulator Sickness To measure simulator sickness, participants were asked to indicate which symptoms, as indicated by Kennedy's 1993 16-point questionnaire, they were currently experiencing after each of the techniques [41]. Simulator sickness seems to be of least concern for Bird (Figure 2-13).

Discussion

This user study suggests that Bird is a usable 3D pointing device concept. Novice users gained proficiency with Bird within minutes. Even with the low-budget data glove, the Bird is competitive with the state of the art in direct object manipulation beyond the natural working volume, as regards its learning curve, enjoyment, and immersion. According to participants, the Bird met the design objective of enabling users *"to move things more freely and somewhat rotate or stack them."* Users reported that they were *"able to manipulate objects much more intuitively and rapidly than with a standard controller."*

Participants enjoyed using the Bird about as much as the most enjoyable technique, Reel. One user commented that the Bird *"feels a little like superpowers."* Regarding Bird, many said *"that one's fun once you get the hang of it."* Although *"this one is a little bit tricky,"* they said, *"the more I used the glove the more I got used to it;" "even though it took some initial time and a little frustration to get familiar," "it gets comfortable pretty fast"* and *"overall it was very convenient and fun."* One said, *"This technique is definitely the most fun to play with."* Meanwhile, while participants enjoyed Reel more on average because the *"task was straight forward"* and they *"felt pretty solid, comfortable and confident,"* still, *"it gets less challenging and could be boring."* Several commented that *"holding buttons and waiting for the effect on the cursor significantly contributes to breaking immersion,"* saying that *"it felt more like...using a game console...in virtual reality...it lessens from that 'realness' factor."* Compared with Bird, Reel *"feels like the most efficient one because I can control the cursor more precisely, but [it is] less exciting."* As for Go-Go, the next most enjoyable technique after Bird, we think that Bird was more enjoyable in part because of its improvement over Go-Go in fatigue.

Since one of Bird's most anticipated improvements on existing techniques was to reduce user exhaustion and address the gorilla arm problem, it is encouraging that Bird ranked well in fatigue. Compared with Go-Go, which was *"a workout"* requiring several users *"to hold [their] arm all the time"* with their other arm, the Bird *"felt like [they] could spend more time during this technique and play around moving objects."* Contrast with Go-Go was especially notable at long ranges. Whereas Go-Go *"gets difficult as the distance between me and the object gets bigger, because I need to be very precise with my movement"* when the arm is at full extension, users generally kept their arms more comfortably near their torsos when using Bird and none commented on arm fatigue. However, several users *"felt a little bit more fatigue in [their] hand[s]"* because *"the gestures necessary...were a bit more for[eign]."* A few said that *"frequently opening and closing"* made their *"hand uncomfortable."* Presumably, *"certain objects at farther distances were more difficult to pick up"* because the participants' fingers were strained opening their hands as widely as possible. In light of these results, we think that Reel outperforms both GoGo and Bird on fatigue metrics because its comfortable button interface does not involve any full-range motor extension. We observe that the calibration procedure used for Bird could be modified easily to use a smaller range of finger motion. This change should place Bird in the same motor regime as Reel, which makes for a promising combination of comfort and controller-less operation.

Users did experience some frustration with Bird. It is hard to disambiguate how much difficulty with the Bird was due to the technique itself, how much was the sticky click aspect of this specific implementation, and how much was due to the data glove.

The sticky click of the Bird created a Midas touch problem[39], as users found that they frequently picked up a different object than they intended. While participants *"definitely like the ability to hold onto one object once you've selected it,"* and *"like it when it feels like the object sticks to the cursor after being selected"* because, *"it gives me more sense of control,"* they were frustrated by the fact that, *"when I was trying to select one item, it would select the previous item instead."* One noted that, *"It would have been nicer if that feature wasn't there."* Since our implementation did not have a way to detect when a user has lost focus on their last item e.g. by moving away, these results are expected in retrospect, but seem like a solvable problem.

Much of the difficulty participants had with the Bird was related to the data glove used to implement it. A few tests were cut short due to hardware difficulties, needing on-the-fly electrical repairs, and some users experienced "vibration" at short distances. Participant comments support that these interruptions probably contributed negatively to, and may account for, reports of user frustration.

The Bird was the only technique which was implemented using hardware which I made myself. The other three all used a standard Oculus Touch controller. I therefore find it to be very promising that the Bird, when implemented using a prototype custom input tool made from less than $100 USD of materials, performed similarly to techniques which all required a commercially available controller. After this proof of concept, I am confident that a concerted expert engineering effort could resolve the issues we faced and further reduce cost.

Clicking with the pointer finger while using the other fingers to control the position of the cursor was very intuitive to some participants, but very difficult for others. The difficulties some users had with *"getting used to using differ[e]nt fingers"* is a serious concern, leading to *"objects...getting clicked while expanding and closing"* the hand, or *"unconsciously mov[ing] our fingers"* which could *"disturb the command."* Wondering, *"Is there a way to curl my finger without curling my other fingers?"*, one subject resorted to *"curling [their] pointer [finger] manually with [their] other hand."*

I see the Bird as a tool suitable for experts in virtual environments, the utility of which does not "bottom out" as its users gain proficiency. 5 of 15 participants independently noted that for them, the Bird took longer to learn, or had more of a learning curve, than the other techniques, but that, by the end of the 2-20 minute task, they felt competent with it. As they became more expert with Bird, they were able to use it with increasing competence. Meanwhile, several participants expressed frustration with the Reel technique, generally the easiest to learn, because it was impossible to use it more quickly and fluidly as they became more familiar with it. I see the Bird as a tool well suited for a future wherein virtual reality is a more commonplace way to interact with information, and which will have many expert users who will become easily frustrated by techniques optimized for less experienced users. The user experience of using the Bird for a longer period of time could be productively explored.

Participants in our study suggested using different fingers to click (which I explore in the second experiment), using a separate device to indicate selection with the other hand, and reversing the direction of the movement of the bird, such that a fully open hand would bring the Bird all the way to the hand.

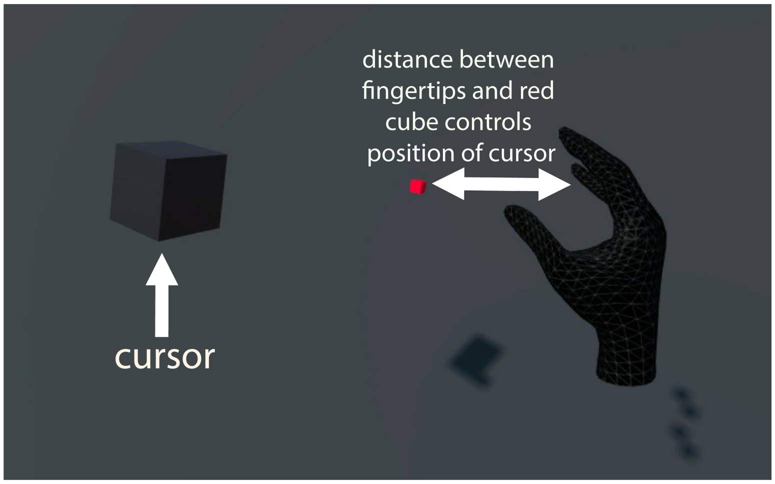

distance between fingertips and red cube controls position of cursor
cursor

Figure 2-14: The second implementation of the Bird, which was used in the user study. The black box is the cursor, and the black hand is a representation of the user's hand. The distance between the Bird and the hand is a function of the distance between the fingertips of the hand and the small red cube in front of the hand.

2.2.5 Prototype 2

Due to the COVID-19 pandemic, further testing of the physical hardware prototype with human subjects became impossible. I therefore reimplemented a second version of the Bird which follows the same principles, but which uses the hand tracking native to the Oculus Quest. The cursor travels along a ray perpendicular to the palm. I calculated the overall openness of the fingers by tracking the distance between each fingertip and a point which was visible to the user directly in front of their hand. We calculate the bentness of the clicking finger, and therefore if a click should occur, in the same way. I then used this measure of openness to determine, each frame, where along the ray the cursor should be. The distance between the Bird and the palm of the user's dominant hand is the total maximum distance the Bird was allowed to travel away from the user's hand minus the extension distance for that frame, where the extension distance was the overall "openness" of the fingers for that frame (calculated based on their distance from a point in front of the hand), multiplied by a scale factor. This scale factor is defined as the maximum distance divided by the range of possible openness values determined in the calibration phase.

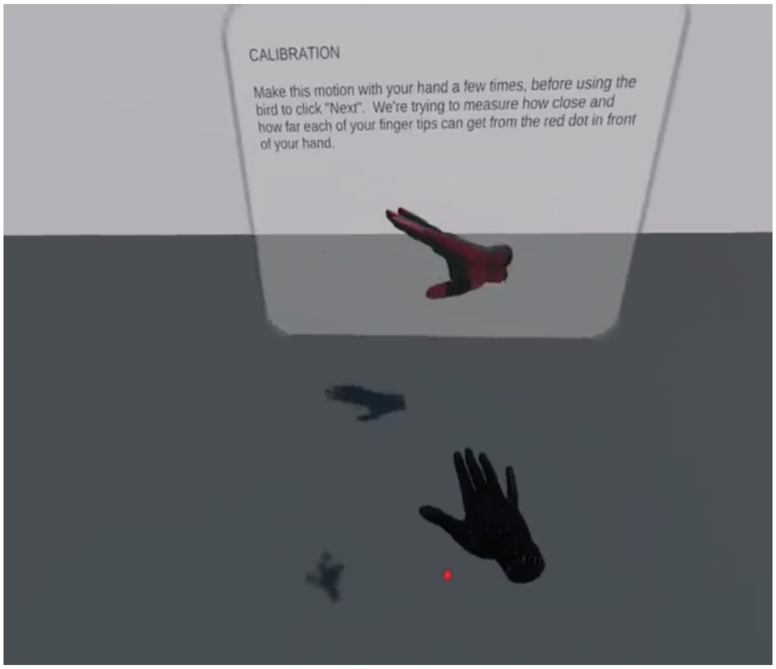
2.2.6 Experiment 2

Due to the COVID-19 pandemic, all experimental activities took place exclusively virtually, with no physical contact between the experimenters and participants. Because the Bird can be implemented on any HMD with hand tracking, I worked only with participants who already owned an Oculus Quest- a specific type of virtual reality headset which has native hand tracking. Participants downloaded the experiment as an application to their personal device. All information regarding the experiment was contained within the application itself, and participants completed the experiment without the experimenter present.

Participants and Apparatus 15 participants (median age 27, standard deviation 10.66 years, 3 women, 10 men, 1 non-binary, 1 prefer not to say, 14 right-handed, 1 left-handed) were recruited. Participants received $30 remuneration. Participants interacted with the task using their own Oculus Quest Head Mounted Display.

Task All information in the experiment was delivered to participants via text located in the same position in the VR scene, occasionally aided by animated 3D models of hands showing them the correct way to move their hand in order to use the technique. On starting the experiment, participants participated in a calibration phase during which we measured the range of motion of their fingers.

Participants were then asked to perform a docking task for each of four experimental conditions for 5 minutes each [50]. In this docking task, they grabbed a red docking sphere, and dragged it to a blue target sphere using the Bird in a room containing 50 white sphere distractor targets. When the docking sphere was successfully dragged to the target sphere, two of the distractor targets became the next docking and target spheres. All spheres were 40cm in diameter and were randomly distributed in 16mx16mx8m space. If a docking sphere was not successfully dragged to a target sphere in 30 seconds, we determined that the participant was either unable to locate the docking or target spheres, or that they were impossible to reach, as the hand tracking on the Oculus Quest often failed in informal testing when trying to grab


CALIBRATION

Make this motion with your hand a few times, before using the bird to click "Next". We're trying to measure how close and how far each of your finger tips can get from the red dot in front of your hand.

Figure 2-15: A screenshot from inside of the experiment. Instructions were provided by text on panels like the one shown, and were occasionally supplemented by a 3D model of a hand demonstrating the desired movement.

spheres directly above the user. In these cases, we randomly chose another pair of spheres to be the docking and target spheres instead.

Participants were allowed to practice the technique and make sure that they understood it after receiving instructions, and could continue on to the formal trial when they felt ready by pressing a start button. Between each condition, participants were asked to remove the headset and fill out the correct section of an accompanying survey, but were not given a specific interval of time to rest.

Conditions I tested two possible fingers for clicking, and two variations on defining which object should be selected, using a 2X2 within-subjects design. Conditions were counter balanced using a Latin square to account for order effects.

The two possible fingers which could be clicked with were the pointer and thumb, because the pinky, ring, and middle fingers are all more difficult to move while holding the other fingers still than the index and thumb. Our two possible variations for defining which object should be selected are both outlined in Steinicke 2006[76]. One, “distance calculation”, always selects the nearest selectable object. The other, “stick-ray”, always selects the object which was last intersected by the ray.

Results I defined a successfully completed task within one trial as one instance of the Bird being used to select the docking sphere, and brought to the target sphere. I defined a failed task as each instance wherein the docking sphere was not selected and brought to the target sphere within 30 seconds. I assumed that, for these tasks, one or the other spheres was located in a position which made it impossible or very difficult to select due to the hand tracking of the Oculus Quest, which tends to become unstable when the hand is not directly in front of the body. Data for both of these measures were pre-processed using an Aligned Rank Transform (ART) to take into account the non-normal distribution.

A repeated-measures ANOVA reveals a significant effect of finger used to click on successfully completed tasks (F1,39 = 15.50839, $p < 0.001$), with index clicking having a higher rate of successfully completed tasks than clicking with the thumb. A repeated-measures ANOVA reveals a significant effect of selection technique used to specify which sphere to select on unsuccessfully completed tasks (F1,39 = 4.86123, $p < 0.05$), with the "sticky-ray" resulting in fewer failed tasks.

Discussion The Bird allows users to manipulate objects far beyond their natural working volume. It does not suffer from the laser pointer dilemma, because the user controls a specific point, rather than a ray. Unlike Jedi ForceExtension, it can translate and rotate objects with no explicit mode switching[21]. Unlike FlexPointer, it does not require the use of both hands [28]. Unlike HOMER, it is a useful tool for both grabbing objects which are far from the user, and bringing them within arm’s reach, and grabbing objects which are nearby, and moving them far outside of arm’s reach [13]. Among existing interaction techniques, it most closely resembles Go-Go [60]. However, Go-Go requires that the user extend their full arm in order to reach beyond arm’s length, whereas the Bird only requires that they extend their fingers, thereby reducing overall user exhaustion and the gorilla arm problem [34]. The Bird

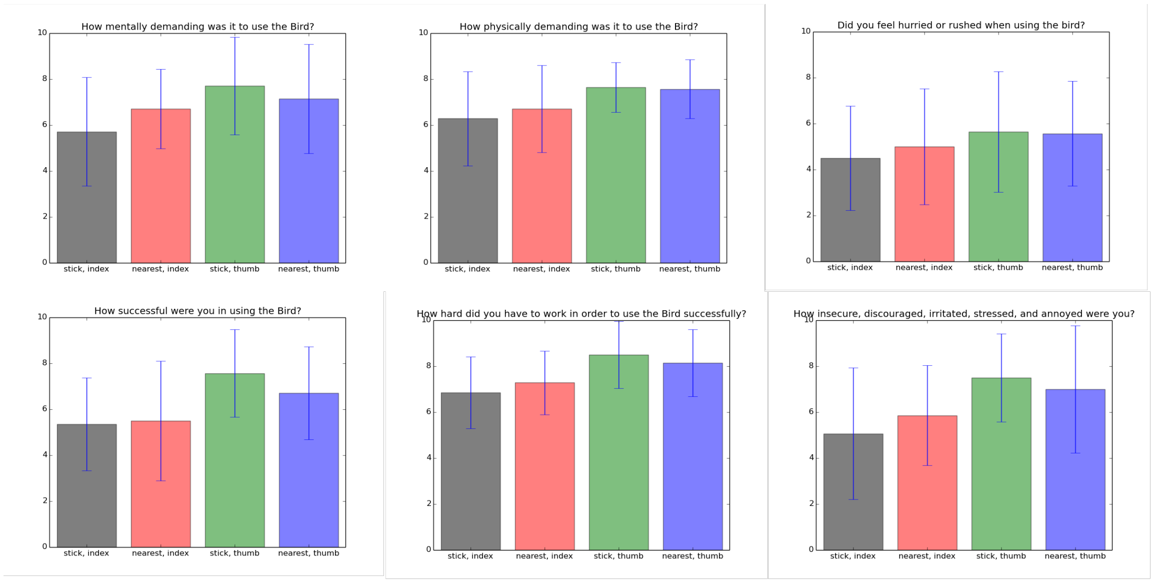


Figure 2-16: Responses to NASA Task Load Index Questionnaire. 10 corresponds to "Very High" or, in the case of "How successful were you in using the Bird?", "Failure". 0 corresponds to "Very Low" or, in the case of "How successful were you in using the Bird?", "Perfect". Error bars represent 1 standard deviation from the mean.

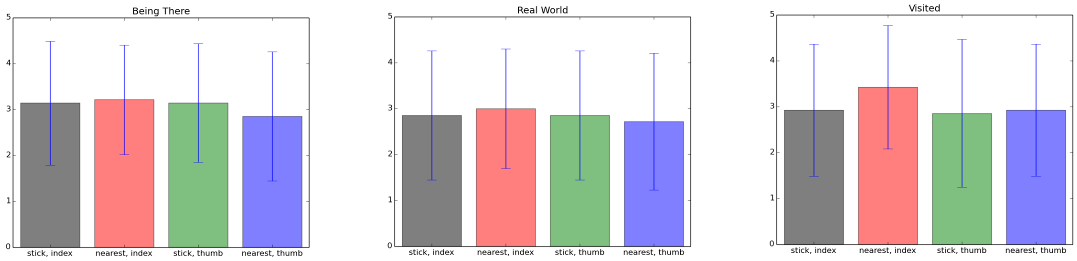


Figure 2-17: Responses to presence questions from Bob G. Witmer's 1998 paper "Measuring Presence in Virtual Environments: A Presence Questionnaire"[81]. Error bars represent 1 standard deviation from the mean.

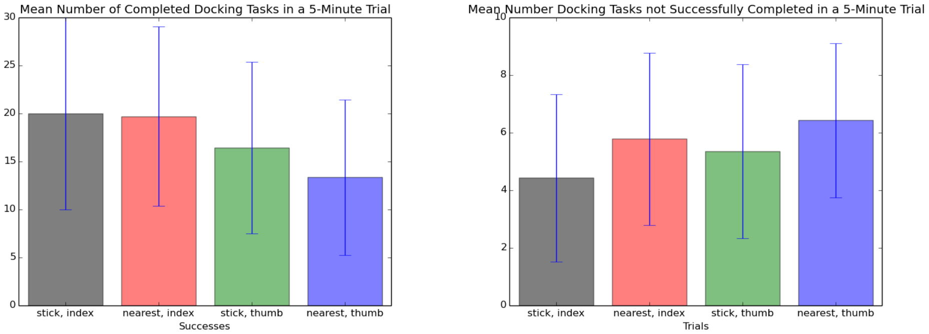


Figure 2-18: Mean number of docking tasks, wherein a red sphere is successfully selected and dragged to a target sphere, completed in a 5 minute trial. Mean number of docking tasks which were not completed within a 30 second window, and were therefore skipped, in a 5 minute trial. Error bars represent 1 standard deviation from the mean.

may even be used with the hand relaxed by the user's side.

In addition to a measurable difference in their ability to complete the docking task, participants had a large number of comments about the experience difference between using the thumb and index finger to click. Participant 6 commented on the increased mental load involved in using the thumb to click, noting that, *"having the thumb involved definitely gets me thinking more"* and *"moving the thumb completely independent of the fingers takes more conscious thought than I would have first thought"*. Other participants described clicking with the thumb as *"very difficult"*(3), *"finicky"*(13), *"less usable"*(1) and *"frustrating"*(10).

It's difficult to isolate to what extent participants' difficulty with variations of Bird were caused by the technique itself, and to what extent they were caused by the technology used to implement the technique. 5 different participants independently noted difficulties related to hand tracking. Participant 5 described it as *"wonky"*, while participant 4 called it *"very shaky"*. Participant 9 noted simply that *"the hand tracking was bad."* Two noted issues related to the instability of the hand tracking when the hands are not clearly visible to the HMD. Participant 1 noted that *"while my hand is near the HMD it sometimes makes random unintended selections"*. Participant 3 said that *"I need to hold my hand in front of the Quest to reliably have hand tracking, which gets straining after a while, taking me out of the experience,"*

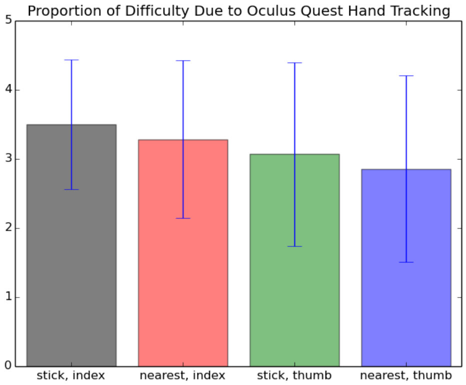


Figure 2-19: Responses to the question "To what extent did your difficulty with the technique stem from the technique, and to what extent did it stem from the hand tracking?" 5 corresponds to "All of my difficulty was with the hand tracking". 0 corresponds to "All of my difficulty was with the technique". Error bars represent 1 standard deviation from the mean.

suggesting that this caused greater fatigue and was detrimental to immersion.

This suggests that the technique may benefit greatly from future improvements to hand-tracking technology. While it was necessary to use the Oculus Quest's hand tracking in order to conduct our remote study during the COVID-19 pandemic, it would be valuable future work to explore implementations of the Bird using either a data glove, such as that discussed in section 3.1, or using an optical tracking system with a larger field of view, such as an Optitrack.

This also suggests that participants were trying to use their hands in ways that involved holding them down or to the side rather than where they are looking. Participant 3 also noted that *"It seemed that every time I turn or briefly look away from my hand it would register as many clicks,"* which suggests that they would prefer to be able to complete an action without having to keep their head oriented continuously towards any specific part of that action. This may have implications for the future integration of gaze input with this technique.

This work suffers from several limitations which are unique and due to the virtual nature of the study. Because I required that users already have their own Oculus Quest, I excluded anyone who did not already own one. I therefore cannot claim that our results are generalizable to all people. Oculus Quest owners are people who own a relatively expensive and new piece of dedicated virtual reality equipment. I can therefore assume that they have more experience with and interest in virtual reality than is standard.

It is notable that the lack of diversity in our results is somewhat different than that which is standard of HCI research conducted at universities, which tests exclusively or almost exclusively with undergraduate students who are disproportionately White. While our participants were not exclusively undergraduate students at the same university, 4 of the 12 who responded to the question had "some college" or an associates degree, and the other 8 all had a bachelors degree or higher. Our participant pool also notably included only 3 women, and 1 non-binary individual. Our population pool was disproportionately but not exclusively White, with 9 of the 12 participants who responded to the question identifying as White (includes Europe, Middle East, North Africa), 4 identifying as Asian (includes East, Southeast, South Asia), 1 identifying as Black or African American, and 1 identifying as Native Hawaiian or other Pacific Islander.

Finally, by being conducted in a hands-off and virtual manner, the experiment has higher ecological validity, but much less control than would have occurred during an experiment conducted in the lab. I have no way of knowing if participants experienced distractions in their physical environments while taking part in the experiment, for example. There was also no opportunity for participants to ask clarifying questions during the experiment, or for the experimenter to notice and correct problems in the participant's understanding of the task or calibration of the Bird.

2.3 Conclusion

I explored a novel tool for interacting with virtual environments, the Bird. It eliminates the need for buttons or a controller, and provides a natural, open-handed interaction option that *"feels like superpowers"* (Participant , experiment 1). With inexpensive hardware, or hand tracking which is now native to some HMDs, the Bird is an accessible interaction to implement. It shifts the focus from selecting objects at specified points to specifying a point anywhere within a 3D volume, which happens to be able to select and manipulate things, and it can do so without any controllers or buttons. The Bird isn't a tool for interacting with a specific selection task. It's a tool for interacting with virtual environments, in much the same way that a mouse isn't a tool for interacting with an application– it's a tool for interacting with all applications. My results suggest that, in future similar designs, clicking using the index finger provides a superior user experience to clicking using the thumb, and a "stick-ray" which always selects the last intersected object may be preferable to selecting the nearest selectable object.

Chapter 3

Menus and 3D Tools

3.1 Related Work

> Note: While the related work section on selection and manipulation techniques for virtual reality is arguably exhaustive, covering all major published instances of point cursors in immersive virtual environments, the related work section on menus and 3D tools only provides a representative overview of the state of the field. This is in part because the previous work on menus in virtual immersive environments is much more extensive than that on selection techniques, and in part because it is more difficult to define specific peers for Pockets, the menu management technique introduced in this section.

3.1.1 Menus

The traditional rectangular, text-based, linear menu has existed for decades. In 1989, Tekla Perry defines the menu as "a list of command options currently available to the computer user" [57]. As early as 1986, [44] proposed embedded menus as an alternative to linear menus, and in 1988, [19] was one of many papers comparing linear menus and pie menus.

3.1.2 Menus in VR

A common way to have an abstract effect on a virtual immersive environment, such as switching applications or changing settings, is to use a 2D menu in a 3D space. These menus can be tethered to a specific location in the environment, to the viewport of the user, or to some part of the body. When tethered to world-space, as a flat panel, they need to be oriented towards the user in order to be usable. A user moving around the space also disrupts this orientation. World space menus can be arrayed in an egocentric fashion or turned to always face the viewer, but this, like screen-space tethered menus, occludes a large portion of the environment.

The ill-suited fit between flat panels with text on them and immersive virtual environments has been understood for longer than the author of this document has been alive. Rudy Darken in 1994 notes that “the direct analogy of two-dimensional menus to three-dimensional applications has not been widely accepted due to complications concerning pointing tasks in virtual space”[25]. He also points out that text on menus in particular is a problem because “text is inherently two-dimensional but is represented in three dimensions” and “[t]he user can move too far away for the text to be legible or might be at some angle from which the text is skewed making it unreadable.” Darken makes this comment immediately before proposing screen-space menus, interacted with via speech, and encouraging “enforcing that execution of inherently two-dimensional tasks take place in two dimensional space,” a point which is directly at odds with the current document.

[66] compared world-tethered and screen-tethered panels of text of varying levels of transparency in a VR history project for K-12 classrooms, and found that world-tethered panels which appear near the user, rather than near the object of interest, and which are semi-transparent were preferable. The preference for semi-transparency arose from the fact that a fully opaque panel large enough to display readable text in a virtual environment occluded a large portion of the screen, but an overly transparent panel was difficult to read.

[67] compared radial and linear menus which could be either “spatial” (tethered to

world space) or “non-diagetic” (tethered to screen space). They found that selection times were faster for radial menus in both world space and screen space conditions, and otherwise found all four conditions to have similar error rates and usability.

Other authors have explored the haptic difficulties involved in making selections from a floating panel. [84] explored the trade-off between placing a 2D menu in the air near the relevant task and lower, but co-located with a physical desk, and found no significant differences between haptic and non-haptic conditions.

In 2014, [45] introduced Finger-Count Menus. Finger-Count menus displayed a number of options on a 2D screen, each labeled with a number. A user selected an item by holding up a number of fingers corresponding to the number of the menu item. [16] released a similar menu for immersive virtual environments in 2001, which attached menu items directly to the fingers.

3.1.3 Proprioceptive Menus

Several previous authors have proposed to tether menu items to various parts of the body. Linear panel menus tethered to the body, most often the hands, have to be small in order to avoid becoming unwieldy. However, tethering proprioceptive menus to parts of the body other than the hands requires either tracking more points of the body, or working out the location of more points of the body using inverse kinematics.

The majority of proprioceptive menus are in some way hand-based. Open Palm Menu is a small, 2D, panel-based menu system which appears in front of the non-dominant palm, which can be interacted with using the fingers of the dominant hand [8]. HandPoseMenu is also a small, 2D, panel-based menu next to the hand, but rather than selecting elements from it by pointing to them with the dominant hand, users select elements from it by holding up a number of fingers corresponding to the positon of the item they would like to select in the list [56]. DigitSpace explored the use of the fingers themselves as a touch interface, rather than a space to tether interfaces near [36].

[79] compared grid-style menus attached to the hand and tethered to world space, selected from using either a hand or head-based pointer, and found that world tethered

menus pointed at with the hand had the lowest error rate, fastest completion time, and lowest task load as measured by the NASA TLX [33].

[51] compared traditional and radial panel menus tethered either to the hand or to world space, and found little difference between conditions. However, they did find hand-tethered radial menus to have statistically significantly lower SUS scores than traditional linear world-tethered menus, which seems to contradict the results of [79].

An excellent example of a proprioceptive menu which is not hand-based is the Tap-Tap Menu[9]. Tap-Tap was a 3D modeling application for virtual reality with menu items stored all over the body– on the arms, legs, hands and abdomen. Menu items were tethered to the user's body by using a Kinect.

3.1.4 3D Menus

[70] explored 3D buttons and switches on panel menus. They focused on the fact that flat panel menus floating in 3D space in VR generally provide no haptic feedback, and referred to these 3D buttons and switches as "pseudo-haptic", because they provide a visual substitute for haptic feedback. They found that, in spite of the design being more distant from mobile or desktop computing designs, "the pseudo-haptic UI performs better in terms of all tested aspects including workload, user experience, motion sickness and immersion."

3.1.5 3D Tools

Prior work has explored the use of 3D tools, rather than 2D menus, as a means of interacting with virtual environments. However, when 3D tools are used, they are still often selected from a 2D menu, for example in [30], which uses an egocentric menu.

Virtual Grasp[83] and TouchTools [32] both allow users to grab 3D objects by miming holding them. Virtual Grasp did so in a virtual environment, but the 3D objects grasped weren't tools– they offered no ability for the user to have an effect upon the environment. TouchTools were for tablets, rather than immersive virtual environments. Because of this, the objects grasped were 2D images of 3D tools. The tools did allow users to have an effect upon their environment– miming a marker would allow users to draw, and miming an eraser would allow them to erase, for example.

A wealth of 3D tools in virtual environments have been used for making music [23][40][85][52][53][29]. This may be because physical instruments are physical 3D tools. Several of these 3D tools are simulations of instruments in virtual environments, such as VRMin, which is a simulated theremin, and Cirque des Bouteilles, which is a simulation of bottles half-full of water [40][85]. Others are more abstract means of representing sound, or the ability to create sound, as 3D objects. Songverse represents sounds and the various effects which can be applied to them as planets [23]. Other virtual musical instruments more directly map individual 3D objects to specific notes [52][29].

I'd like to highlight Cirque des Bouteilles [85] in particular as an example of a 3D tool, and virtual environment design which effectively disregards precedent drawn from desktop and mobile computing in favor of something more creative and more fundamentally 3D. Cirque des Bouteilles is a virtual experience of "making music by blowing on bottles". Cirque des Bouteilles also does not include a single traditional panel-based menu. Each bottle is a self-contained 3D tool. A bottle can be played by pointing at it and blowing. Every bottle is labeled with the name of the note it will play. Every bottle can be picked up, moved around, and reorganized in the virtual environment to facilitate playing them, in the right order and using them to make music. The type of bottle and amount of liquid in it can be changed using a set of very evocative gestures directed specifically at that bottle. Bottles can be destroyed by breaking them ("hitting them from above"), and bottles can be created by dragging them from a "special self-replicating bottle". There is no menu with a "generate new bottle" button labeled with text. Instead, interactions with bottles draw from our understanding of how a glass bottle would work in the physical world– they can be picked up, moved around, broken, and labeled with their contents. They also have a correct orientation baked into the physical affordances of a physical object– an upside-down bottle would pour out all of the liquid[55]. Bottles in Cirque des Bouteilles also step beyond what is possible using a real bottle in the physical world– the level of water in a bottle can be controlled using a simple gesture, rather than pouring, the glass shards won't cut you, and infinite bottles can be created. This is a precedent I try to follow in my own design of 3D tools.

3.1.6 Inventory Systems in VR

While menu systems for virtual reality have been studied extensively, inventory systems in virtual reality are a largely under-explored space. I define inventory systems in 1.1 as *A menu system wherein contents are user-assigned.*. [80], which is discussed further in the next section, claims to "Compare Two Inventory Design Concepts", but does not allow users to assign objects to specific item slots in either inventory design concept, thus causing them to not meet the author's definition of "inventory". Cmentowski et al. creates a taxonomy of inventory systems for virtual reality games, and compares inventory formats [22]. These systems were a flat grid, in which all items were represented as 2D images, a set of virtual drawers, in which items were represented as equally-scaled models, and a Magnetic Surface, in which items were simply attached to the surface of the inventory at their standard size. None of the systems involved hardware beyond the headset and controllers standard for an HMD. In all 3 inventory systems explored, the system and all objects within it were visible at all times. This is interesting in that "One major concern with all designs was the occlusion problem between environment and inventory. As VR inventories have to be placed within a 3D scenario, they can easily obstruct important content or appear behind other objects" [22].

3.1.7 Toolbelts in VR

Many examples exist of virtual reality experiences which involve a toolbelt [20][58][30][80]. In all instances I found in the literature, the toolbelt was visually represented to the user in the virtual environment, placed at the waist of the user based on the position of the HMD, and was a menu rather than an inventory– that is, the creator of the application determined which items went where in the toolbelt, and the user had no opportunity to change this configuration. [80] compared this variety of toolbelt to a hand-tethered menu and found no significant differences between designs in spatial presence or usability.

3.1.8 Virtual Pockets

The term “virtual pockets” in reference to user interface design can be found exactly twice in the literature. The first is a wearable computing paper from 2000 called “Virtual Pockets”[46]. Virtual Pockets proposes that the desktop metaphor is ill-suited for wearable computing. Instead, it suggests drawing metaphors from clothing. Virtual Pockets allow users to store information in specific locations tethered to their clothing using touch-sensing fabrics. It was implemented, but the prototype "was not supposed to be fully featured *Virtual Pockets*, but rather a working version for the basic functionality" [46]. “Virtual pockets” turn up again five years later as the subject of “Virtual Pockets in Virtual Reality”[38]. “Virtual Pockets for Virtual Reality” proposes a five option proprioceptive menu system, and demonstrates it using a drawing task. The five “pockets” are arrayed around the user’s head, and their contents (“tube”, “line”, “eraser”, “magnet” and “color”) are all determined by the creator, rather than the user. “Virtual Pockets for Virtual Reality” does not incorporate additional physical hardware to represent its pockets, and instead places each “virtual pocket” at a set distance from the location of the HMD.

3.2 Pockets

3.2.1 Concept

Pockets are a means of interacting with virtual environments which minimize the necessity of 2D menus in 3D space. Pockets are a highly modular, user-assigned menu based on physical buttons. The hardware involved in Pockets is any set of buttons

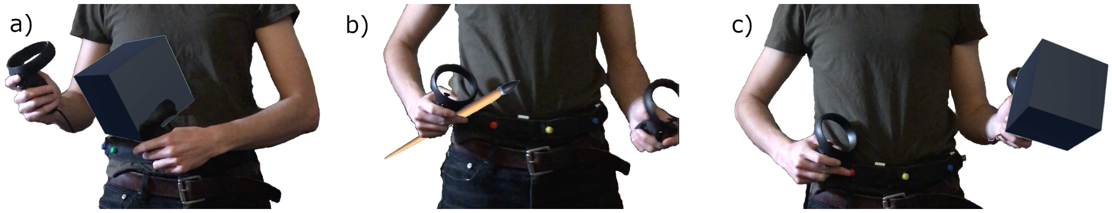


Figure 3-1: a) Pushing a button while holding a virtual object assigns the object to a Pocket. b) Different objects can be assigned to each Pocket. c) Objects can be retrieved by pushing the corresponding button. Because Pockets use physical hardware, and because we return objects to the dominant hand, we do not need to track their location.

which are able to convey that they have been pressed. This could hypothetically be a keyboard, the buttons on controllers, or more custom hardware. Our current implementation consists of six buttons worn around the waist, to draw from the real-world metaphor of a toolbelt.

The use of a toolbelt metaphor is inspired by Mark Mine's 1997 paper "Moving Objects in Space", as discussed in 1.2. A tool belt is a means of carrying objects used to modify and interact with the world around you, generally carried by construction workers around a job site, though often used in other professions and activities. Pockets are somewhat more casual than a tool belt. We generally, if we wear clothing which includes them, carry fewer things in them, but carry those things most of the time. What we carry is similar for each person, but has slight interpersonal variation. For example, most people carry their phone in their pocket, but only some people carry chapstick, and what pocket they keep these objects in varies from person to person. What we carry in our Pockets also changes over time, or depending on the activities we intend to engage in. For example, I now never leave my house without an N-95 mask in my right back pocket, a fact which was never true prior to the pandemic.

When a button is pressed while holding an object in the dominant hand, the object is deactivated and assigned to that Pocket. Metaphorically, it has been stored "in" the Pocket. If a button is pushed which is already assigned to an object, the object is retrieved from the Pocket, and instantiated at the location of the dominant hand. Because the button is aware of when it has been pushed, and because the hand is tracked, there is no need to track the location of the button in this implementation.

By user-assigned, I mean that Pockets start empty, and users add objects to them as they need to. This means that users never have to learn a pre-made menu. Instead, they develop patterns of keeping objects in the most convenient Pocket for any particular activity, similar to the way we might keep our phones in the same pocket of our pants, unless we are running, in which case we may move them to a more stable arm-band.

By using physical buttons worn on the body, we can take advantage of proprioception and passive haptic feedback to avoid representing Pockets visually in the HMD. A user knows the approximate location of buttons on the body due to having seen them when putting them on, and will likely develop muscle memory over time, in the same way that skilled typists do not need to look at their keyboard when typing. A physical button provides the user with passive tactile feedback to help them understand when they have located and pressed a button.

When worn on the waist, this peripheral hardware serves as a comfortable location to also mount additional cameras, to extend hand-tracking to areas which cannot be seen from the HMD, such as below or otherwise outside of the field of vision. It also provides a more comfortable location to attach additional compute power than worn directly on the HMD.

This relates to the discussion of privacy in 1.4.3. I hope that, should Pockets be adopted in a broader way, the affordances offered by physical hardware worn on the waist, such as extended hand tracking, will encourage future designers looking to reduce the weight of HMDs to move computation to peripheral hardware worn somewhere more comfortable on the body, rather than to move it off of the device.

3.2.2 Hardware and Implementation

For our initial prototype, we implemented Pockets using 6 buttons of 12mm diameter, spaced at a distance of 8cm apart. These buttons were wired to a Feather M0 board, which then communicated with the larger application over USB. This set of buttons was sewn into 1.5" webbing, and attached to an adjustable belt using velcro straps.

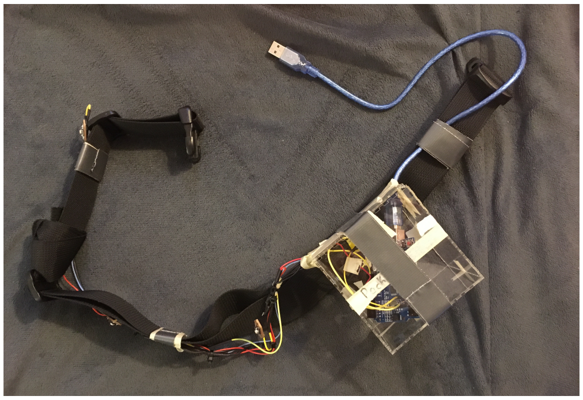


Figure 3-2: Initial prototype. While it worked, it wasn't of an adjustable width, and was not robust enough to survive a remote study.

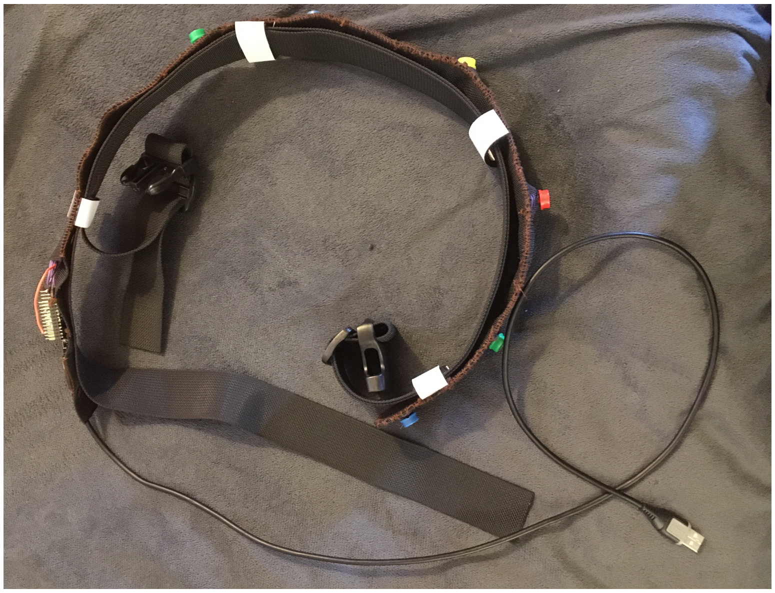


Figure 3-3: Second prototype, which was used for both studies.

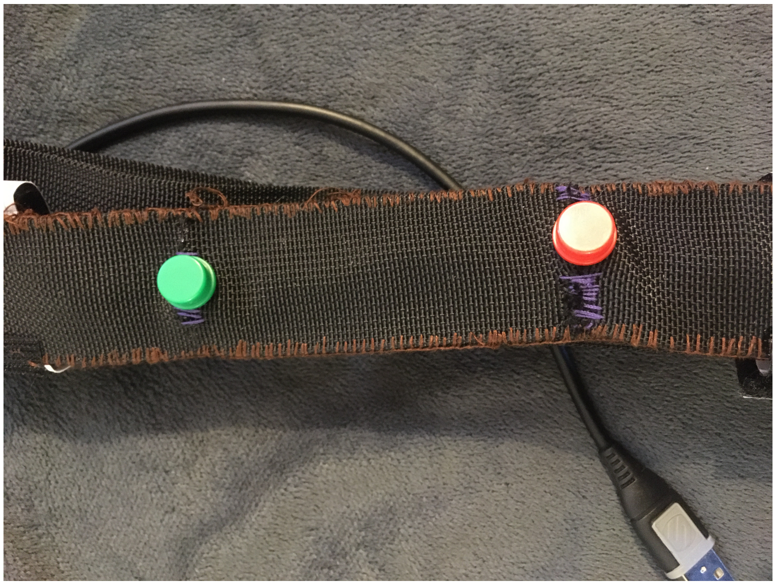


Figure 3-4: Close up of buttons on the tool belt.

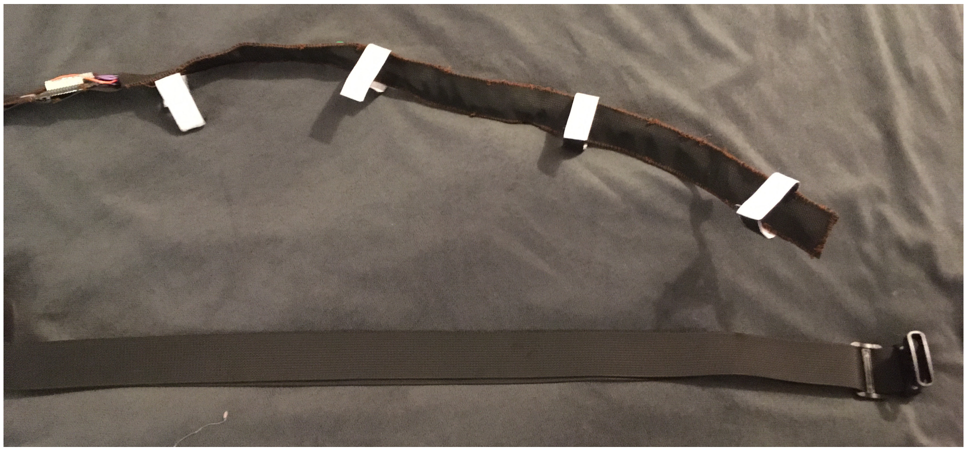


Figure 3-5: The tool belt separates into two parts, held together with Velcro straps. This allows the belt the be extremely adjustable, but the buttons to be a fixed distance apart.

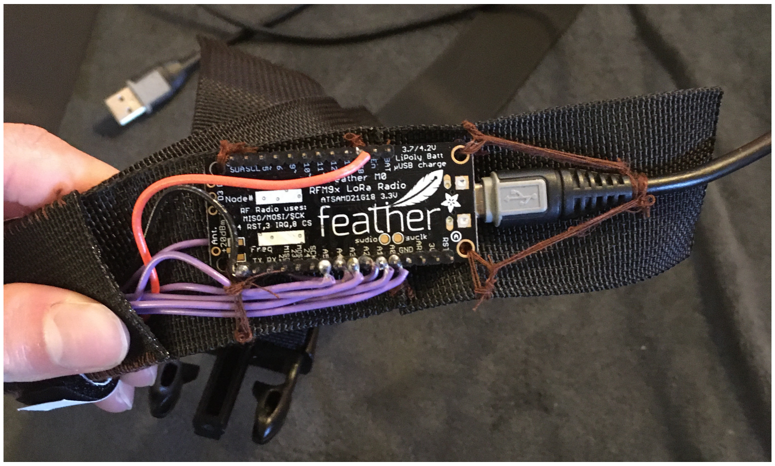


Figure 3-6: Connections to the microcontroller exposed for easier repairs. The connection between the belt and USB cable is tied in place to strain relive it, and reduce the likelihood of a participant disconnecting it during a study.

The application was developed in Unity 2018.4.

Because the first experiment was remote, I needed to make it possible for Pockets to be able to be plugged into a Windows computer and immediately work for users with minimal technical knowledge. This required some interesting work related to COM ports. Our experiment was designed to work with Windows machines, which assign a COM port number to a new device in a way that seems to be at random, and sometimes change that COM port number while the device is still plugged in. I therefore needed to write scripts which could both automatically connect to the correct COM port, and could reconnect if the COM port changed while the experiment was running. Future readers of this document are encouraged to explore the code. All code related to this experiment can be found at the link listed in appendixA.

Ideally, Pockets would be paired with bare-hand interaction. This would leave a user's hands available in order to more easily find the buttons. It also prevents the redundancy involved in using a physical toolbelt of buttons while holding controllers which, themselves, have buttons. However, in these early prototypes, this was not possible. The first prototype communicates with the headset and larger application over USB, and the protocol involved in communication over USB with the Oculus Quest, the only commercially available headset to offer hand tracking, was too chal-

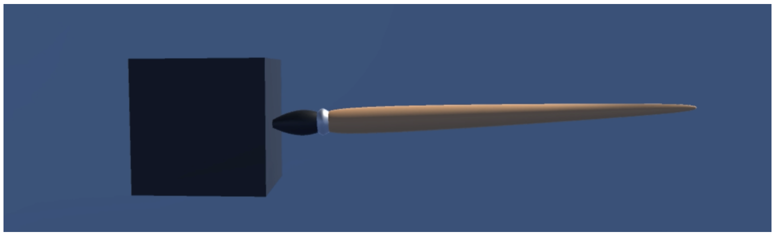


Figure 3-7: Paintbrush.

lenging to implement on a limited time scale. I see exploration of Pockets paired with bare-hand interaction as essential future work.

3.2.3 Experiment 1

As pandemic restrictions lifted, I ran a pilot study (n=6, 1 female, 2 left-handed), to better understand how people interact with Pockets. We explored the following research questions:

1. How many pockets do participants actually use, and what is the distribution of button clicks across pockets?

2. How many "mistakes" do we see, if we infer that any instance where someone takes something out of a pocket and immediately puts it back to be a mistake?

3. Do we see a reduction in "mistakes" over time?

4. Is the waist an uncomfortable place to wear pockets?

This study was conducted virtually, but with hardware which was delivered to participants. To follow the precedent set by [38] and [9], I implemented a simple sculpting task. Participants were asked to recreate a series of simple sculptures made from 3D primitives using a set of 3D tools which we implemented for use in this study. These 3D tools were as follows:

Paint Brush A tool for painting and pointing. It can be used to move the sliders on all generators, to point to specific colors on the Blob Generator, and to change any paintable object to the color of the Paint Brush.

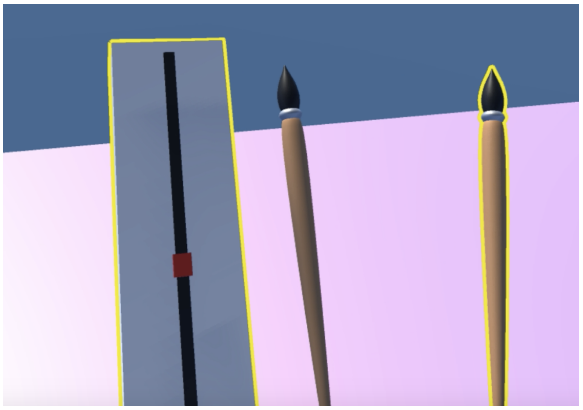


Figure 3-8: Brush generator.

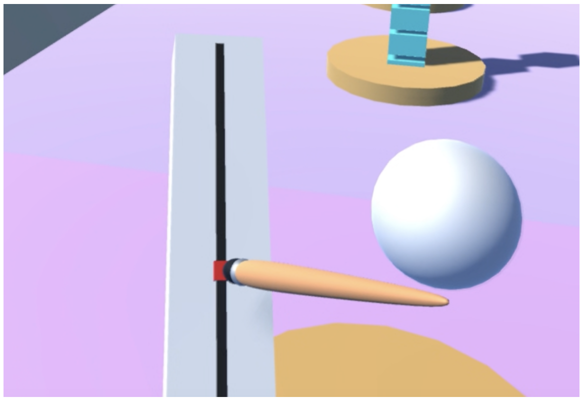


Figure 3-9: Sphere generator.

Paint Brush Generator A tool for generating paint brushes. Paint brushes can be used to move the slider, which controls the size of the paint brush to be generated. Paint brushes can be instantiated by removing them from the generator with the hands.

Cube Generator A tool for generating cubes. It works exactly like the Paint Brush Generator, but generates cubes instead. Cubes are paintable objects.

Sphere Generator A tool for generating spheres. It works exactly like the Cube and Paint Brush Generators, but generates spheres instead. Spheres are paintable objects.

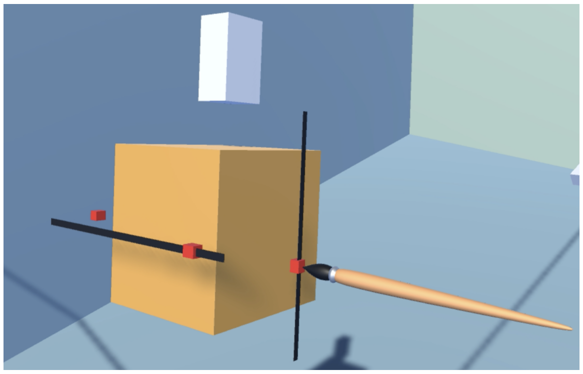


Figure 3-10: 3-Axis Generator.

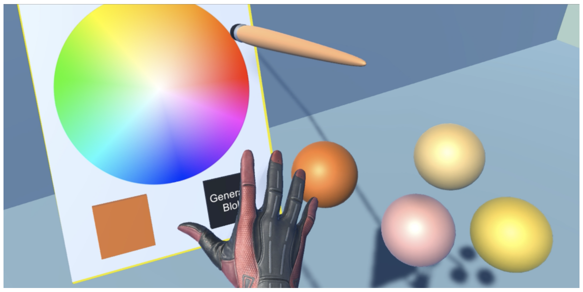


Figure 3-11: Blob generator and generated blobs.

3-Axis Generator A tool for generating cuboids. Paint brushes can be used to move sliders which control the length, width and height of the cuboid. The position of the slider on the generator represents which dimension it controls. Objects can be instantiated by removing them from the generator with the hands. Cuboids are paintable objects.

Blob A blob is an abstract unit of color, which operate like a blob of paint. The color of a paintbrush can be changed to the color of a blob by touching the end of the brush to the blob.

Blob Generator The blob generator creates blobs. The paintbrush can be pointed at the color wheel to specify a color, which can be previewed in the cube on the bottom. If the color is satisfactory, a blob of that color can be generated next to the generator by pressing the “Generate Blob” button with either the hand or paint brush.

Figure 3-12: Trash can.

Trash Can The trash can could be used to throw objects away. Paintable objects, blobs, and brushes touched to the top of the trash can are deleted. It does not throw away object generators, or other essential elements of the room.

Participants were introduced to each of the 3D tools in a tutorial room. They were instructed on how to use each, and that all 3D tools, along with all other 3D objects which could be picked up and interacted with, could be placed in Pockets. Then, participants were moved to the main study.

Participants could move around both the main study space and the tutorial room using the standard teleportation mechanism found in the SteamVR SDK. If participants spent more than 60 minutes in the study, they were prompted to exit the experiment, and move onto the survey. The time limit of 60 minutes was chosen in order to avoid fatigue effects. We asked participants to record audio, and speak aloud during the experiment. Participants were also asked to fill out a survey at the end, and we logged the following data:

1. The type of object, and the unique id of that object, time, and pocket number for every time an object was put into or taken out of a pocket

2. The time and pocket number every time a button on the toolbelt was pressed

3. The amount of time spent in the tutorial phase

4. The amount of time spent in the experiment overall

5. The location of all objects in the scene, sampled every 30 seconds

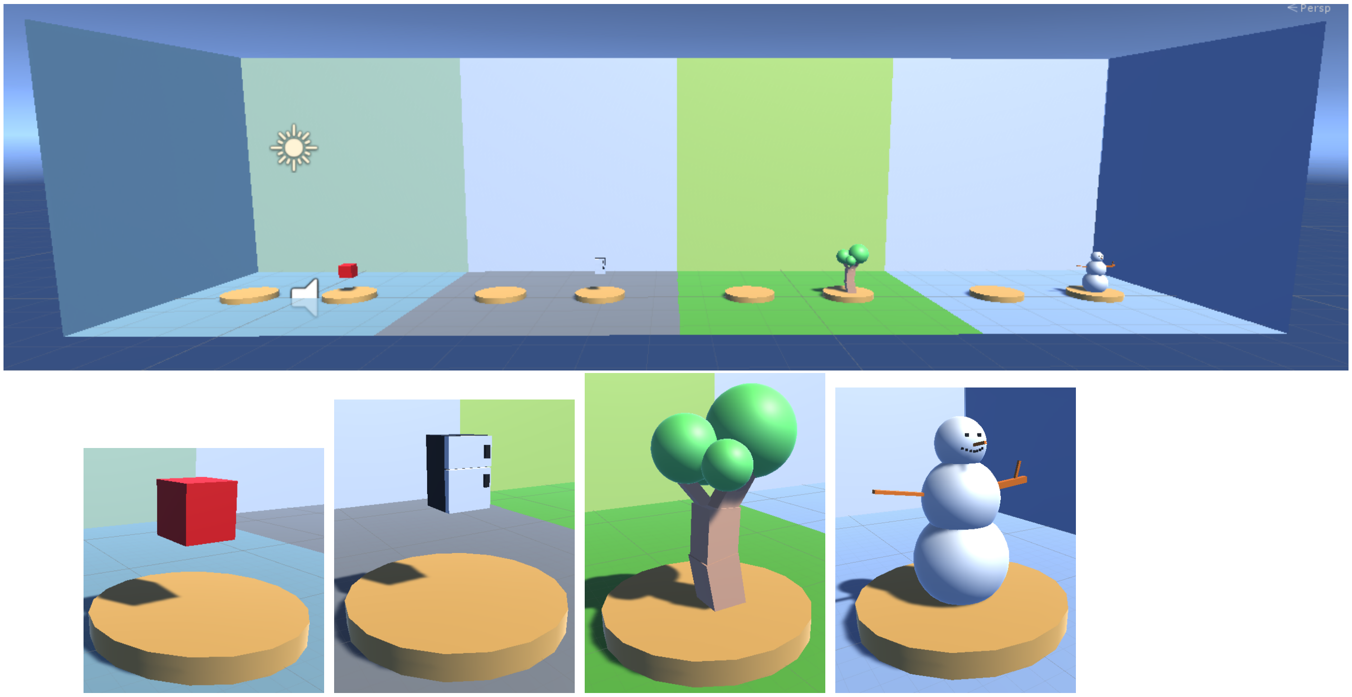


Figure 3-13: Study environment, and closer views of the objects which participants were asked to recreate using the 3D tools.

Results

> Note: Data from participants 1 and 4-7 was missing or incomplete, which is why there is data from participants 8, 9, 10, and 11 in an n= 6 study.

Participants spent an average of 17.5 minutes in the tutorial phase, with a standard deviation of 9.9 minutes, and an average of 53.3 minutes, with a standard deviation of 13 minutes in the application overall.

Based on survey feedback and audio recording transcripts, using the paintbrush as a pointing tool was not intuitive. Participant 1 *"was confused why I needed to use the paintbrushes to adjust sliders"*. Participant 3 wanted *"just to be able to grab the sliders with my hand"* because *"I don't always remember that I need to use the brush tip to control sliders, so I end up having to find my brush pretty frequently"*. Participant 9 immediately understood that the paint brush could be used to move the slider on the paintbrush generator, but then generalized that this meant that a cube needed to be used to control the cube generator slider, and a sphere needed to be used to control the sphere generator slider. When this didn't work, they wondered

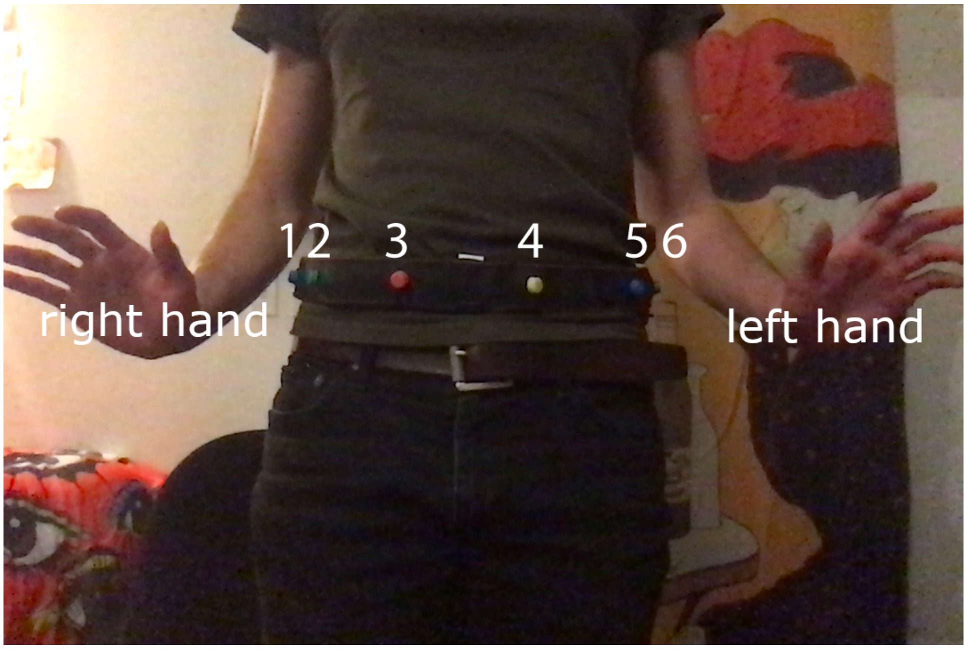


Figure 3-14: Hardware used in both Pockets studies, labeled with button numbers.

if *"maybe I have to do it with the pointer finger"*.

In general, participants (3, 4, 8, 9, 10) were in agreement that the waist was a good location for Pockets, although participant 10 suggested *"it could possibly be on one side like a pager instead of having buttons across your whole front."* Participants 8 and 2 suggested that it might have been more comfortable worn lower on the hips, like a normal belt, and participants 4 and 8 suggested wearing it across the chest *"like a bandolier"* as a potentially interesting alternative. Participant 4 also noted that *"Pockets on the thighs + wrists would be sweet"*, and participant 8 also suggested the lapel as a good place for one additional Pocket, which would be easy to differentiate.

Several participants experienced problems with the buttons of the belt. Participants 3 and 8 both had button caps fall off of the belt, which left them still usable, but harder to find. Participants 2 and 8 both accidentally hit buttons while kneeling down at some point during the experiment. While participant 8 said that the size was *"fine"*, participants 1, and 3 agreed that *"bigger would be easier"*(P1). Participant 3 suggested that *"two or three centimeters"* or *"touch panels woven with conductive thread"* would have worked better because *"something I could reliably hit with the heel of my palm would have been nice"*. Participant 2 appreciated the overall clickiness of

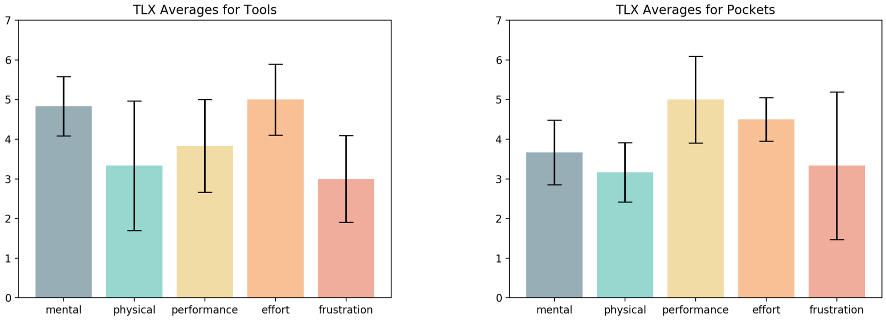


Figure 3-15: Responses to NASA Task Load Index Survey for 3D Tools and Pockets.

the buttons, because it was *"simple to find with your fingers and obvious when you'd pushed it, from the click."*

While the quantitative data suggests that 3D tools were very mentally demanding, our qualitative data suggests that they were also fun. Several participants were particularly excited about the fact that the paint brushes could be made in different sizes, even though this feature largely lacked a specific purpose. Participant 3 pointed out that they *"can make little tiny toothpick paint brushes"*, and that they used a *"gigantic"* paintbrush for the majority of the task, even though it was likely less usable. Participant 10 commented, in reference to a paint brush, *"a small paint brush. Oh my god, it's so tiny."* Participants also liked the infinite nature of the generators. Participant 10 noted that they *"should not be making the great brush kingdom,"* and generated a total of 20 paint brushes. Participant 8 also *"loved"* the trash can, and appreciated that they *"can sweep the trash can around to collect stuff."*

Participants 9 and 11 thought that 6 was the correct number of Pockets, and all other participants agreed that either 4 or 5 buttons would have been preferable, because *"four is about the same number of pockets I use on a pair of irl pants"* (P1). Participant 2 thought that they *"could manage 5 effortlessly–right, left, center, rightish but not right, and leftish but not left"*, but also pointed out that *"for this task i don't think i would have wanted fewer buttons because I needed all of them."* Participant 10 suggested that *"Keeping track of 6 pockets was on the edge of what I could do,"* and that *"when storing things for a long time (over 5min) it was easy to forget*

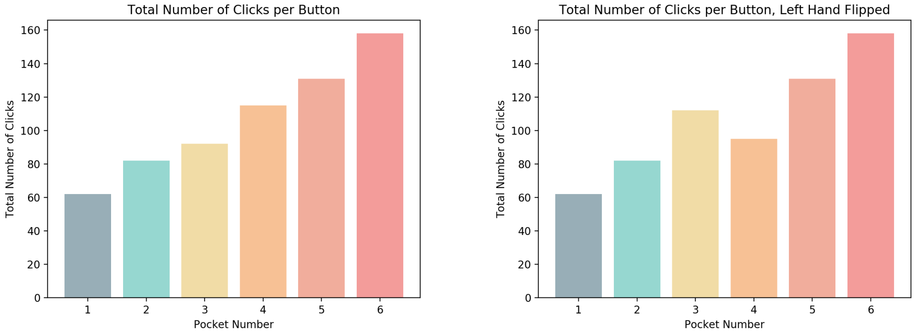


Figure 3-16: Left: Total number instances of a user putting something into a Pocket or taking something out of a Pocket across all users. Right: Total number of instances of a user putting something into or taking something out of a Pocket, but with data inverted for left handed users, such that data from Pocket 6 is counted as Pocket 1, etc.

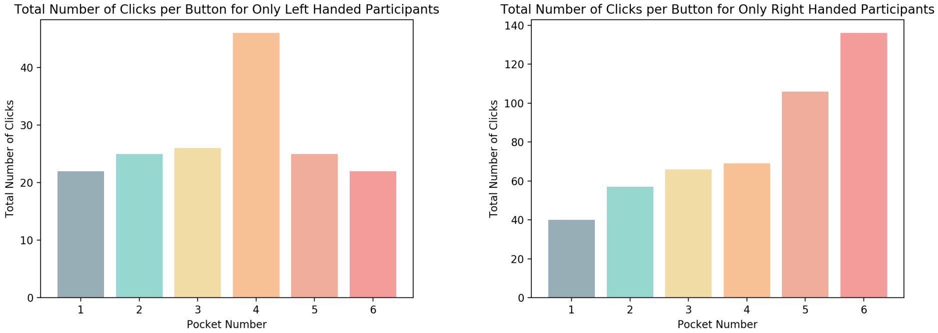


Figure 3-17: Left: Total number instances of a user putting something into a Pocket or taking something out of a Pocket across all left handed users. Right: Total number of instances of a user putting something into or taking something out of a Pocket for all right handed users.

which pocket it was in." This suggests, in short-term use like this study, a connection between Pockets and working memory[10].

All participants from whom we had log data (P2, P3, P8) used all Pockets a minimum of 2 times, and a maximum of 62 times. A Shapiro-Wilk normality test of all Pockets and residuals found them to be normally distributed, and a one-way ANOVA showed a statistically significant difference in usage between Pockets (F(2, 34)=7.3454, $p<.05$). This suggests a systemic preference for Pockets on the left side of the body. This may be because a user needs to be holding an object in their dominant hand in order to put it into a Pocket. This would suggest that users are

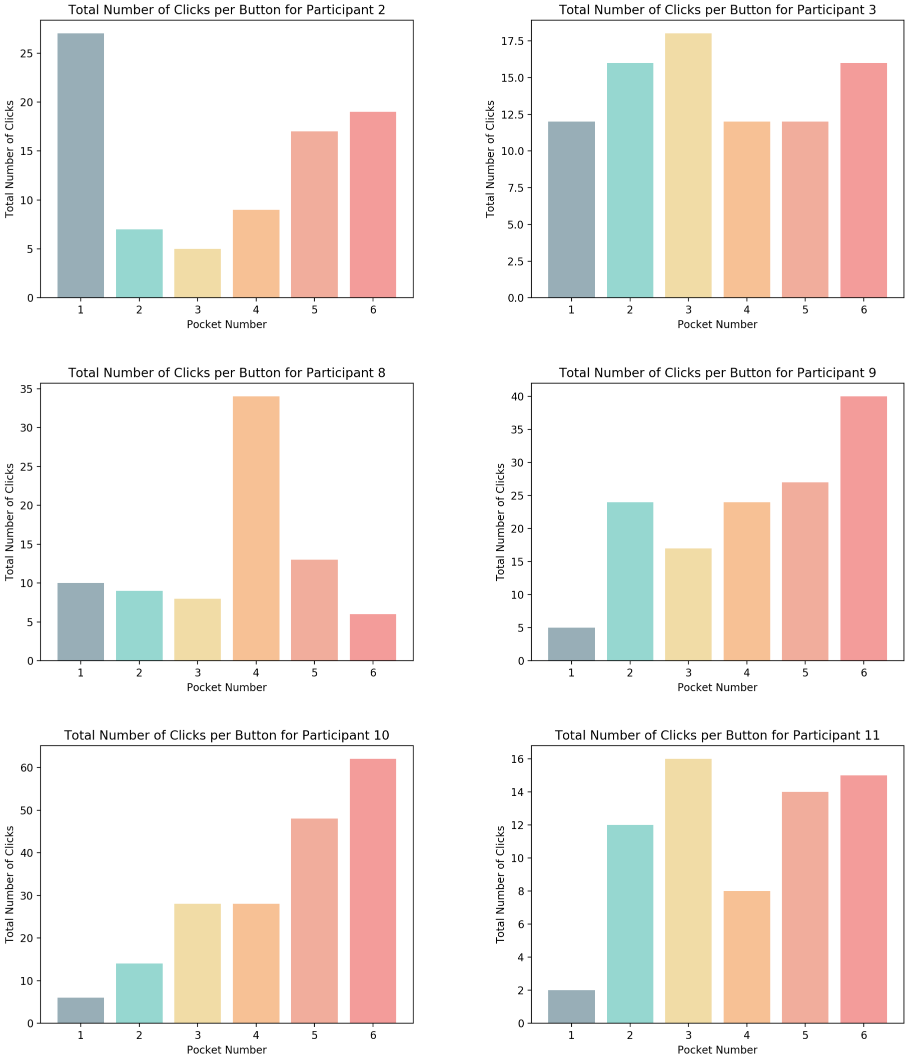


Figure 3-18: Total number instances of a user putting something into a Pocket or taking something out of a Pocket for each user.

pressing buttons with the non-dominant hand while holding objects in the dominant hand. Interestingly, flipping the data for left-handed participants weakens rather than strengthening the effect (F(2, 34)=6.3067, p<.05). This may simply be due to statistical noise, given our sample size of 6.

We can see from the graphs that both participant 2 and participant 8 primarily used one Pocket. For participant 8, this was the middle button mentioned in the transcript as being used for the "orange paintbrush". For participant 2, this was used exclusively for the color selector.

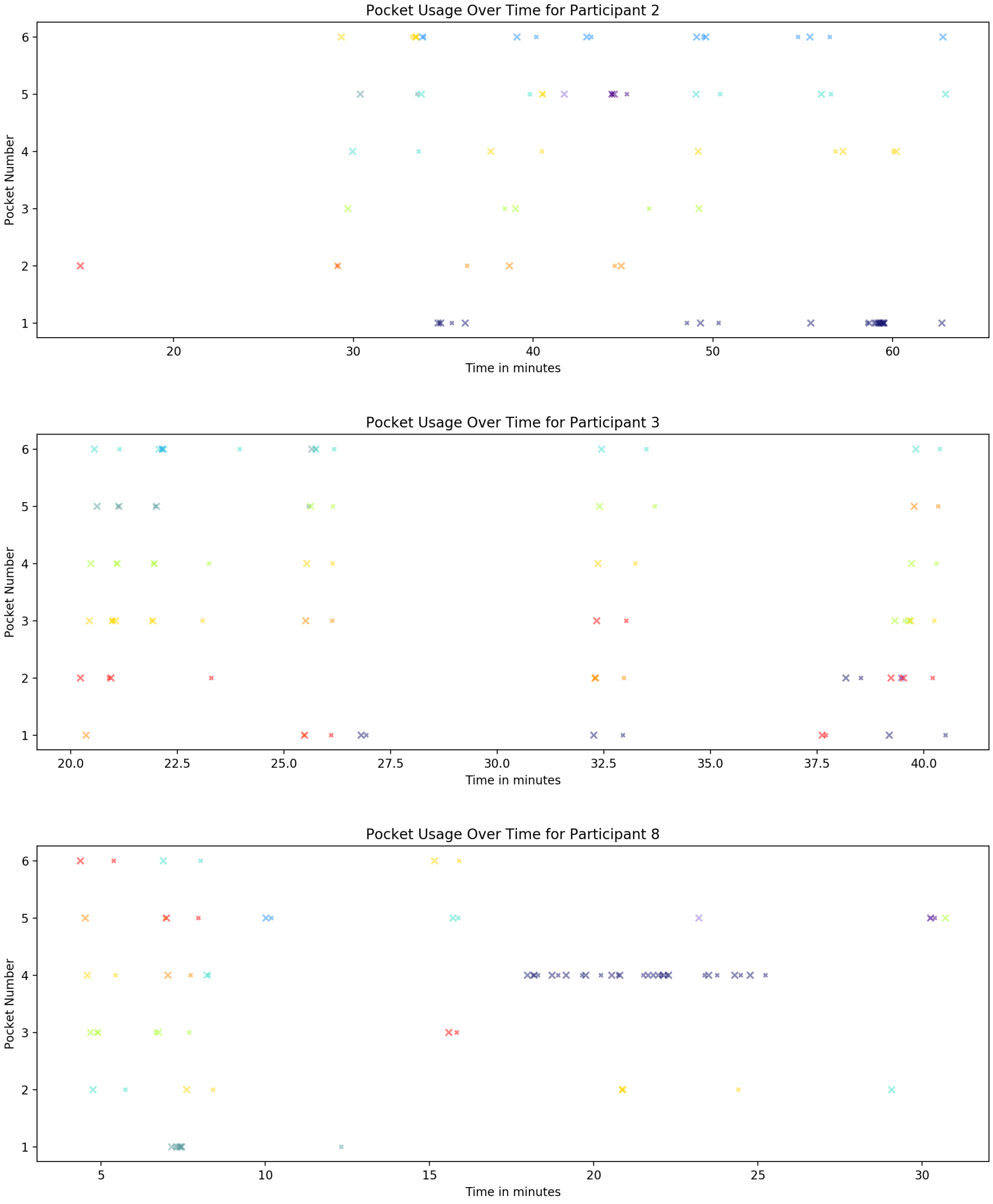


Figure 3-19: All instances of an object being taken out of or put into a Pocket, per user for participants 2, 3, and 8. Larger markers are an object being put into a Pocket, and smaller markers are an object being taken out. Markers of the same color represent the same object.

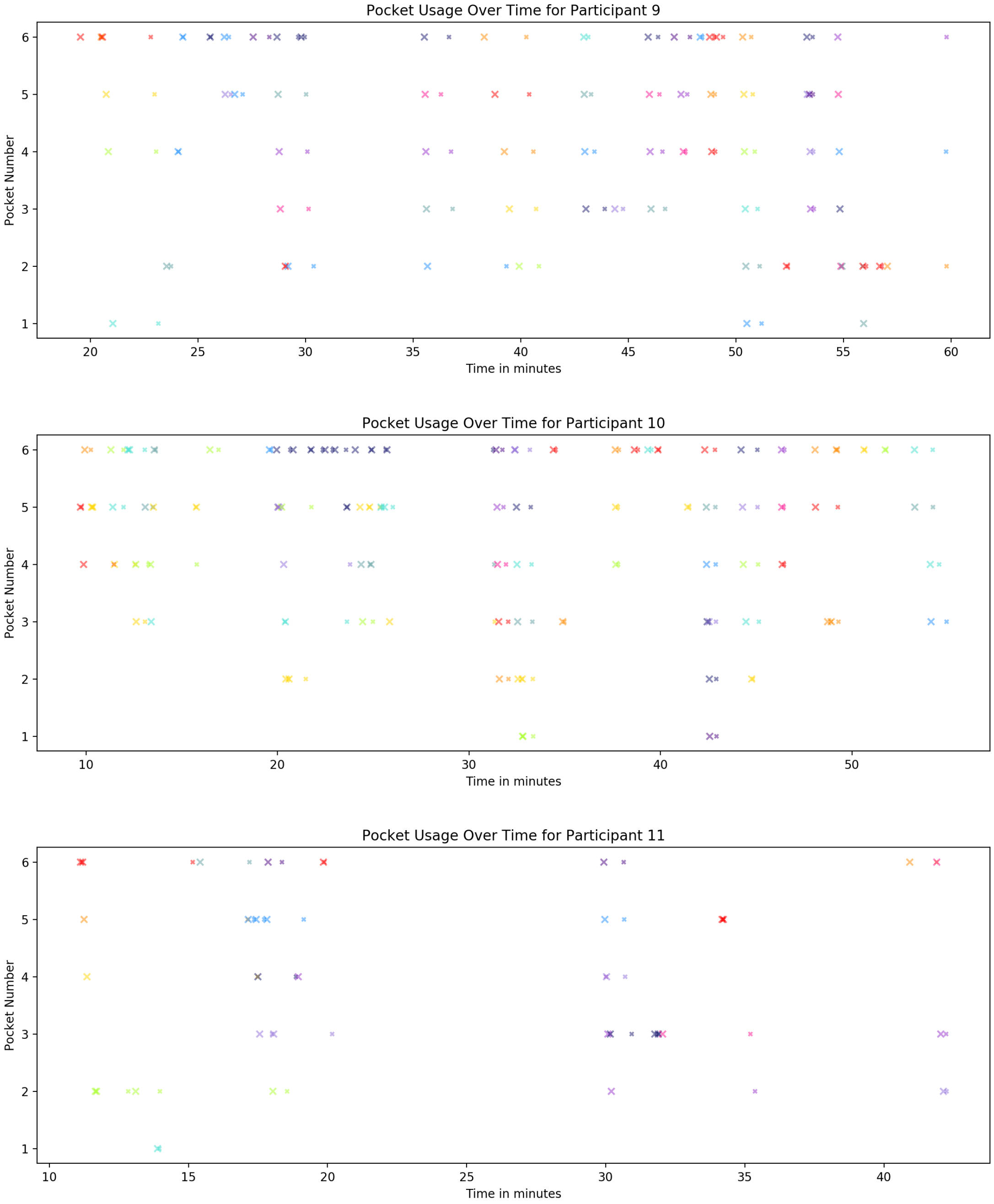


Figure 3-20: All instances of an object being taken out of or put into a Pocket, per user for participants 9, 10, and 11. Larger markers are an object being put into a Pocket, and smaller markers are an object being taken out. Markers of the same color represent the same object.

We can see from the above graphs that all participants initially placed different objects into each of the 6 Pockets, and were relatively consistent about keeping the same objects in the same Pockets for at least a few minutes. Participant 2, who mentioned creating a “system” of “where objects go” in the survey, largely placed objects into the same Pocket they’d been taken out of, but would occasionally switch where an object “goes”.

Participants seem to have struggled to keep track of what was in each Pocket early on, but gradually developed systems for themselves over the course of the study. Participant 2 found it to be “harder at first” to remember which object was in which Pocket, but “but pretty quickly got used to my own system.” They were “glad I made a system, that is, places where things "go," because otherwise I would have had a very hard time remembering.” Participant 3 noted later that “it’s getting more intuitive to use the pockets.”

Participant 8 returned particularly thorough audio and survey data, which created an interesting window into understanding how they used Pockets, and the system they developed for themselves over the hour or so they spent in the experiment. Early on, they found it to be “Easier to navigate by just teleporting around and retrieving objects from where I left them rather than trying to remember which pocket I put things in.” This was because “every time I put stuff in pockets even if it was just a matter of seconds ago when I moved to another location and need to retrieve them. I ended up just hitting all of the pocket buttons to just empty the contents so I can just see.” However, they later noted that they “might be getting the hang of actually making use of Pockets,” and mentioned that they were habitually keeping a paint brush of a specific color in the “front center Pocket”.

Later, they realized that they were “finding it easier to remember what’s in which pocket if it’s an instance of an object instead of a generator.” They noted that, generally, “having a manifestation about an object is easier to remember. Which you know, isn’t that crazy if you think about it for a second. I guess that’s kind of how brains work.” In the survey, they clarified that “When the pockets held abstract objects (color picker, brush generator) I kept forgetting what was where. When a pocket held a specific instance of an object (ORANGE brush) it was very easy to remember.” They also noted that “Once I needed to have multiple paintbrushes of different colors, the utility of having pockets became pretty clear!”

While I wasn’t able to determine “mistakes” from instances of participants taking an object out of a Pocket and very quickly putting it back again as easily as I would have expected, it is possible to visually identify how often someone removed all of the objects from all Pockets in quick succession in order to find something. This also suggests instances of a participant forgetting which Pocket an object is in (rather than forgetting what is in a specific Pocket).

Participants 1 and 3 said that remembering what was in each pocket was “difficult”, and participant 2 described it as “medium difficult”. When participants struggled to remember what was in each Pocket, it resulted in a pattern of largely keeping objects out and visible in 3D space, loading them all into Pockets when it was necessary to carry them to another location, and emptying all Pockets upon arrival. We can see this in participants 3, 9, and 10 especially clearly. Participant 10 “would forget what was in them sometimes and have to empty them out to see what I had.” Participant 8 frequently emptied all of the Pockets “just empty the contents so I can just see.” Participants 3 also “found myself loading everything up then unloading it into a pile and then sorting it, rather than keeping everything in a pocket and using it as needed” because they “sometimes forgot which pocket I stored objects in”

Several participants suggested that a visual indication of the contents of Pockets would have been useful. Participant 3 suggested that “some kind of reference might be useful,” such as “an inventory icon”, “a Minecraft inventory belt” or “a task bar or some sort of UI element that tells me which pockets have what in them, or which ones are filled”. Participant 9 suggested that “being able to see them somehow would be helpful to not have to store it in memory, but I ended up just leaving everything out so it was all available”. Participant 10 found it to be “Hard to remember what I put in what pocket,” but said that “if there was an indicator it would have been fine”. Participant 3 also acknowledged that “maybe that like totally defeats the purpose of what you’re trying to do.”

We therefore explored the trade-off between the increased occlusion of visually representing the contents of Pockets in the virtual environment and the difficulty remembering the location of items which were not visually represented in the environment in experiment 2.

Participants in general seemed to like the idea. Participant 8 said that it would be “fun to see the UI evolve a little more,” because they “think it could become actually really quite convenient to use.” Participant 9 called it a “cool idea!” and said that it was “fun to get to actually use it.” Participant 3 said that it was “awesome” and that they felt “like I’m in the future”.

3.2.4 Experiment 2

Our second study was conducted in person. Due to the ongoing COVID 19 pandemic, this influenced our study in notable ways. First, our participant pool was limited to MIT students who were actively allowed to access the building, which means that our sample is composed exclusively of graduate students. Second, participants were required to wear a mask during the duration of the study, which caused occasional fogging of the headset. On occasions when the headset became too foggy to effectively see, participants were asked to remove the headset, and the lenses were wiped down by an experimenter, before resuming the study.

Our second study was otherwise identical to our first, save for three intentional changes. First, based on P3 (Study 1)’s comment that the blobs and spheres were "identical", and that it would help for the blobs to be "more blobby", blobs were changed from spheres to a more irregular shape.

Second, modifications were made to pointing tools. Based on feedback from study 1 that participants "kept wanting to move the sliders with my fingers instead of a brush, especially if I didn’t have a brush nearby" (P3), we added a second means of pointing at sliders and the color picker. This second means consisted of a cuboid attached to each palm.

Finally, we explored two different means of visualizing the contents of the Pockets, explained below.

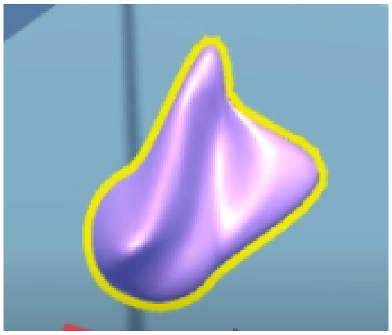


Figure 3-21: Blob used in experiment 2.

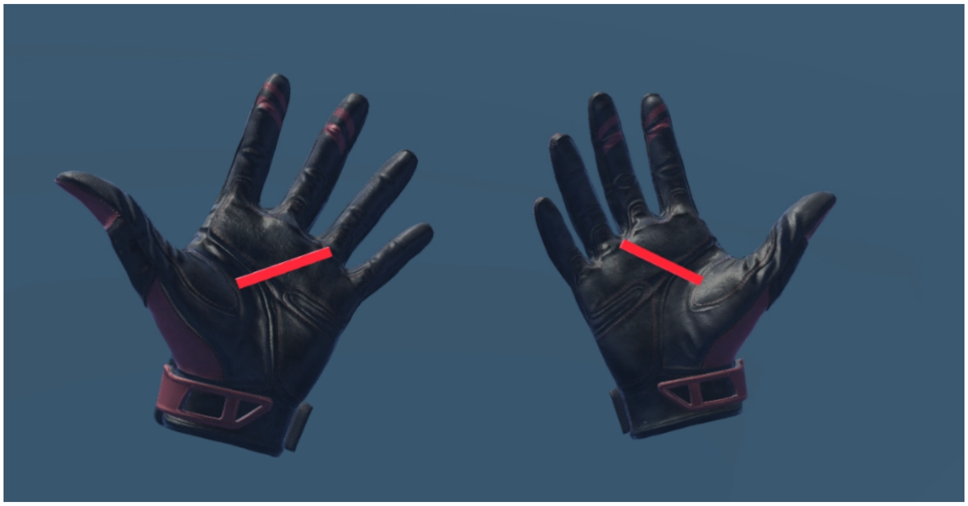


Figure 3-22: Hands with pointing sticks, which could be used as an alternative to paint brushes for interacting with sliders.

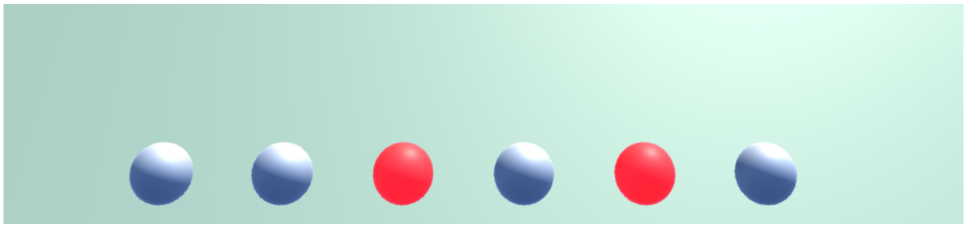


Figure 3-23: Visual representation of Pockets at the bottom of the field of view in the Lights condition. Red spheres represent occupied Pockets, and white spheres represent empty ones.

Conditions

Based on user feedback from our first study, we ran a between-subjects study with three conditions which varied the means of visualizing the contents of Pockets

Control This condition was the same as our pilot study. The contents of Pockets were not visible to participants, and Pockets were not visually represented in any way inside of the headset.

Lights In this condition, each Pocket was represented by a small white sphere always visible at the bottom of the field of view, as part of a heads-up display. When a Pocket was occupied, the corresponding sphere would turn (light up) red.

Icons The icons condition included the same heads-up display as the Lights condition. However, in addition to representing that a Pocket was occupied, the object placed into the Pocket was scaled to half of its size, and stuck into the Pocket. Its rotation stayed the same as the angle it was at when it was stored.

While it would have been more ideal to visually represent Pockets co-located with their physical counterparts, this proved to be challenging, for reasons discussed in section 4.2. Muhammad Imtiaz's 2020 thesis suggests that screen-space displays in virtual reality causes more simulator sickness than world-tethered displays [7]. Comparing two conditions with screen-space visualizations and a control created the opportunity to replicate this finding.

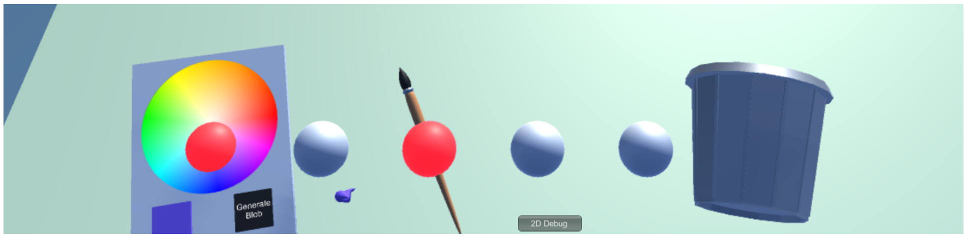


Figure 3-24: Visual representation of Pockets at the bottom of the field of view in the Icons condition. Red spheres represent occupied Pockets, and white spheres represent empty ones. Because objects were scaled by a constant, and were therefore all different sizes, information about the size of the object was still obvious to the participant, but large objects in Pockets could obscure the associated light, and small objects could disappear entirely in lights.

Hypotheses

1. Participants will start to keep the same tools in the same pockets over time

   (a) H0:Users are equally likely to put a tool back into the pocket they got it out of as they are to put it into any other available pocket.

   (b) H1: Users are more likely to put a tool back into the pocket they got it out of than they are to put it back into another available pocket.

2. People will keep things in Pockets more over time

   (a) H0: The number of Pockets occupied by objects will be evenly distributed across time spent in the study.

   (b) H1: The number of Pockets occupied by objects will be higher earlier in the study.

3. Participants will feel more simulator sick in conditions visual representations of Pockets

   (a) H0: There will be the same number of symptoms of simulator sickness indicated in the survey for participants in the Lights and Icons conditions.

   (b) H1: There will be a greater number of symptoms of simulator sickness indicated in the survey for participants in the Lights and Icons conditions.

4. Emptying all Pockets (to look for what is in them) will occur most often in the Control condition, less often in the Lights condition, and least often in the Icons condition

   (a) H0: Events of emptying out all Pockets will happen equally often in all conditions

   (b) H1: Events of emptying out all Pockets will happen most often for participants in the Control condition, and least often for participants in the Icons condition

Results

Across a 90 minute VR study, it's possible for one of any number of things to go wrong at any point in time. Various hardware, software, and administrative errors which occurred during this study caused incomplete or otherwise altered data, as detailed below:

Participant	Condition	Notes
1	Control	No problems
2	Lights	No problems
3	Icons	Tracking lost after 40 minutes
4	Control	No problems
5	Lights	No problems
6	Icons	No problems
7	Control	Left handed participant given right handed version of study
8	Lights	No problems
9	Icons	No problems
10	Control	No problems
11	Lights	No problems
12	Icons	No problems
13	Control	Accidentally exited program early after 25 minutes
14	Control	No problems
15	Icons	Button 5 broke during the study
16	Icons	No problems

Because of these problems, participants 3, 7, 13, and 15 are removed from certain types of analysis. Additionally, participants 14-16 were put in the Control and Icons conditions, rather than evenly distributed across all 3 conditions, to ensure that at least 4 participants completed all conditions with no problems which could affect the data.

Participants spent an average of 15.3 minutes in the tutorial phase (sd = 6.1) and an average of 49.6 minutes in both the tutorial and task (sd = 12.5).

The meaning of both the Lights and Icons made immediate sense to most participants in the Icons and Lights conditions as soon as they put objects into a Pocket, and were generally well received. Participant 6 (Icons condition) noted specifically that "The inventory bar was a great addition." However, while participants generally liked the inclusion of Icons, response to the specific implementation was more mixed.

Participant 6 said, " it was just a tiny bit too low, and when objects went to the bar they appeared to maintain orientation, which made things like generators a bit hard to quickly identify." Participant 12 said, "the visuals were a little crowded," which meant that they sometimes "ended up just pressing them all just to get everything out."

Users in both the Lights and Icons conditions struggled to map visual representations of Pockets to the physical buttons without being able to see them. Participant 2 said that “I could find them generally, but I never really trusted which one I was touching (e.g. 3rd from the left or 4th from the left, etc.) without physically counting across,” but also that, “maybe that would come more intuitively with more practice.” Participant 3 said that, "It was hard to associate the physical place of buttons to the visualization." They also "had to count from the left most button," and "almost never got it right." Participant 6 said that, " It took me a bit of time to build muscle memory for where each of the physical buttons was, but by the end of the exercise I’d mostly figured out where all six lived." Participant 12 similarly noted that buttons towards the ends were easier to find. They said that, "The two center most buttons are difficult to identify. In comparison the two left and the two right-most buttons are easy to "recover.""

Participants in Control and Lights conditions struggled to keep track of what was in each Pocket. Participants 7 (Control), 10 (Control), 11 (Lights), and 13 (Control) all explicitly mentioned forgetting what was in Pockets. Participants 7 and 13 both noted that this frequently led to pressing a button thinking that the Pocket was empty, and accidentally removing its contents, which, according to participant 7, was "pretty annoying".

Several participants in the Control and Lights conditions also described the Icons condition, or ideas similar to the Icons condition, without being informed of it. Participant 4 suggested "Giving the user feedback as to where the tools are being in stored." Participant 7 said that they "need like a something that tells me where stuff is in my Pockets," and suggested a "preview" if they "look down or something" or "if I push one of the random buttons on the controller." Participant 13 expressed a

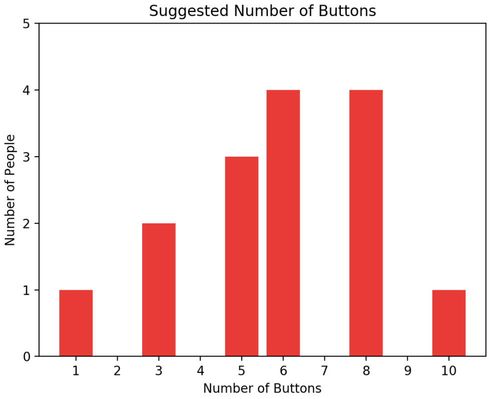


Figure 3-25: Optimal number of buttons, according to participants.

preference for a, "preview of what's in the pocket." Participants 8 and 11 were both much more specific. Participant 8 suggested, "a little icon over the bulbs to reminder yourself what you put in there." Participant 11 said that they "kind of wish like when I put a cube in one of my Pockets the little representative thing was now a cube," or "maybe even just hovering above it, you know a small representation of what's in it."

Participants 6 (Icons condition) and 15 (Icons condition) pointed out that visual representations of Pockets would be more useful if they were tethered to the waist than as a heads-up-display. Participant 14 (Control condition) wanted to "just visually see my pockets," preferably represented as "little avatars of what was in each pocket hovering around in an inner-tube fashion around me." Participant 15 noted specifically that it would help " if the pocket was tracked."

Hardware In general, it was too easy to accidentally activate buttons, but participants would like to still be able to activate buttons with elbows, arms, etc. intentionally. Fewer other protrusions on the belt (such as the exposed microcontroller) would make it easier to find the buttons with fingers. Different textures would have made them easier to differentiate.

Almost all participants accidentally triggered Pockets when bending or crouching, due to either fat or clothing on the torso, elbows, or the controllers bumping into them. This suggests that either the buttons themselves should be redesigned to be less exposed, or the location of the buttons should be changed. Participants suggested a sash (P10, P12), "upper chest / lapels (Star Trek communicator style)" (P6), or "shoulders (like epaulettes)" (P6) as potential alternative locations. Participants disagreed on the optimal number of Pockets (ranging from 1 to 12).

Several participants suggested that varying the shapes and textures of buttons would make them easier to distinguish. Participants 4 and 7 both specified that "unique shapes" would be helpful. Participant 3 noted that, "if their texture was different to touch, that would make everything easier." Participant 7 suggested that "features on the buttons" would help, which participant 14 suggested, "some kind of braille-like system."

Hypotheses

Simulator Sickness All participants experienced between 0 and 5 of the possible 16 symptoms on the Kennedy Simulator Sickness Questionnaire [41]. While the average number of simulator sickness symptoms was lower for the Control condition (2.66) than the Lights condition (3.0), the number of symptoms of simulator sickness was very unexpectedly lowest for the Icons condition (1.40). While this might be partially attributable to participants 3 and 13, who exited the study after 40 and 25 minutes due to various hardware and software failures, and were both in the Icons condition, and spent less time in the experience. If we remove them both from the data, the mean for the Icons condition still only rises to 1.75 symptoms. This suggests that representing Pockets visually as a heads-up-display does not cause greater simulator sickness in participants.

Pockets are for Carrying Things Perhaps the most interesting result of being able to watch 16 participants use Pockets in person was the ability to see that many participants did not think of them or use them in exactly the way that I anticipated. While I conceptualized Pockets as primarily a replacement for menus, participants tended to see them as a more general tool for carrying things. For example, participant 10 (Control condition) said, as soon as they understood how to use the Pockets

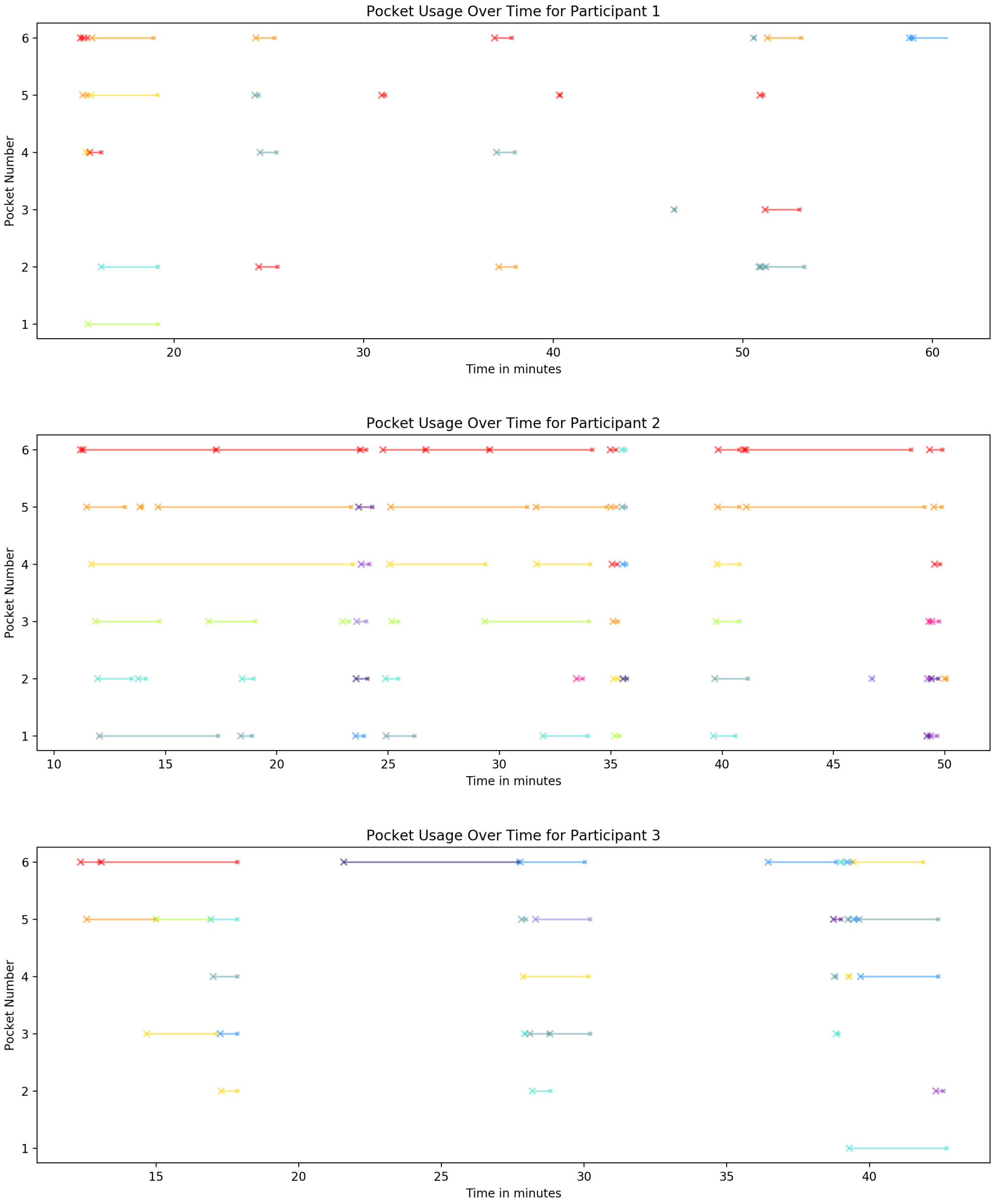


Figure 3-26: All instances of an object being taken out of or put into a Pocket, per user for participants 1, 2, and 3. Larger markers are an object being put into a Pocket, and smaller markers are an object being taken out. Markers of the same color represent the same object. Lines represent time during which the object was in the Pocket.

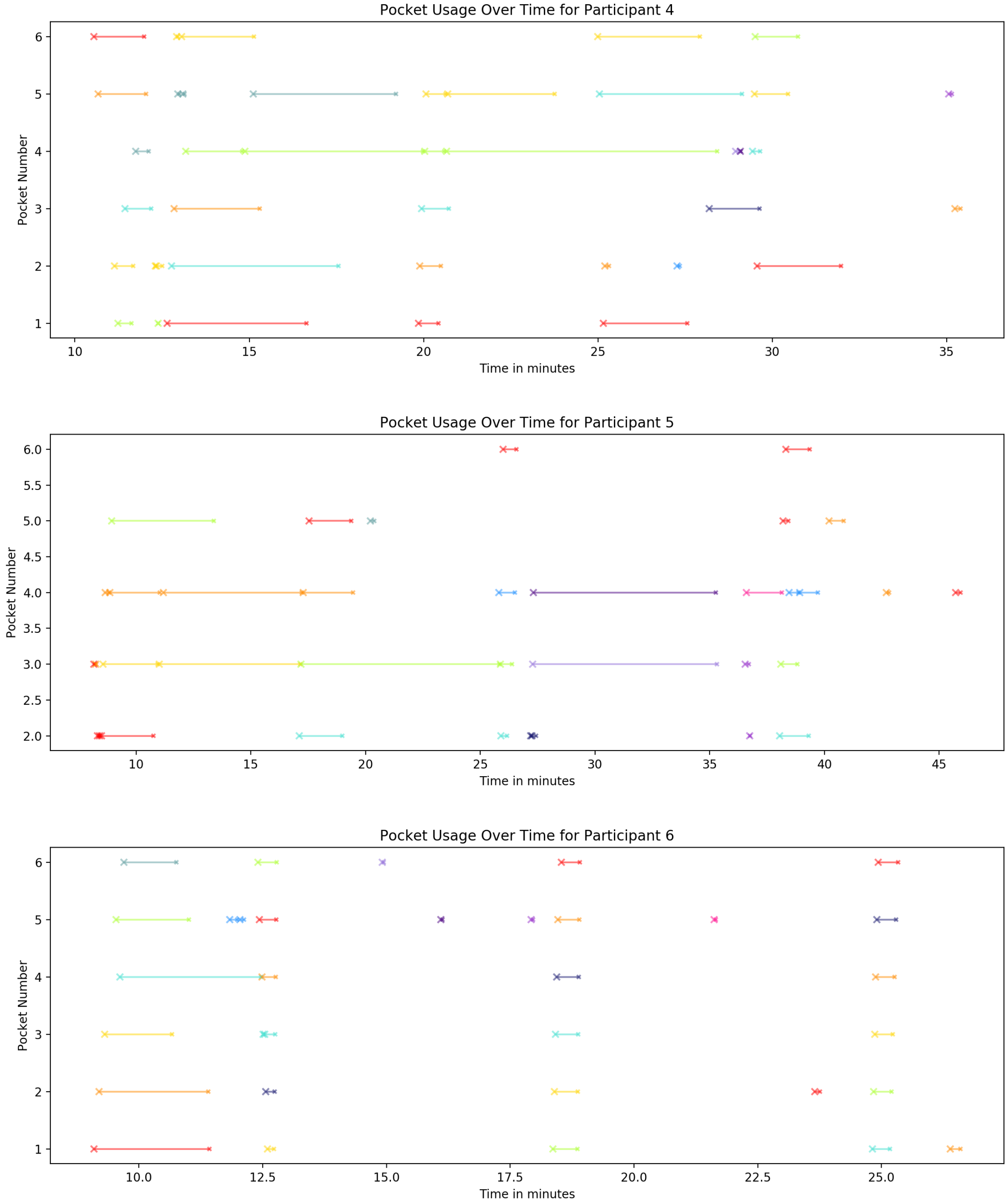


Figure 3-27: All instances of an object being taken out of or put into a Pocket, per user for participants 4, 5, and 6. Larger markers are an object being put into a Pocket, and smaller markers are an object being taken out. Markers of the same color represent the same object. Lines represent time during which the object was in the Pocket.

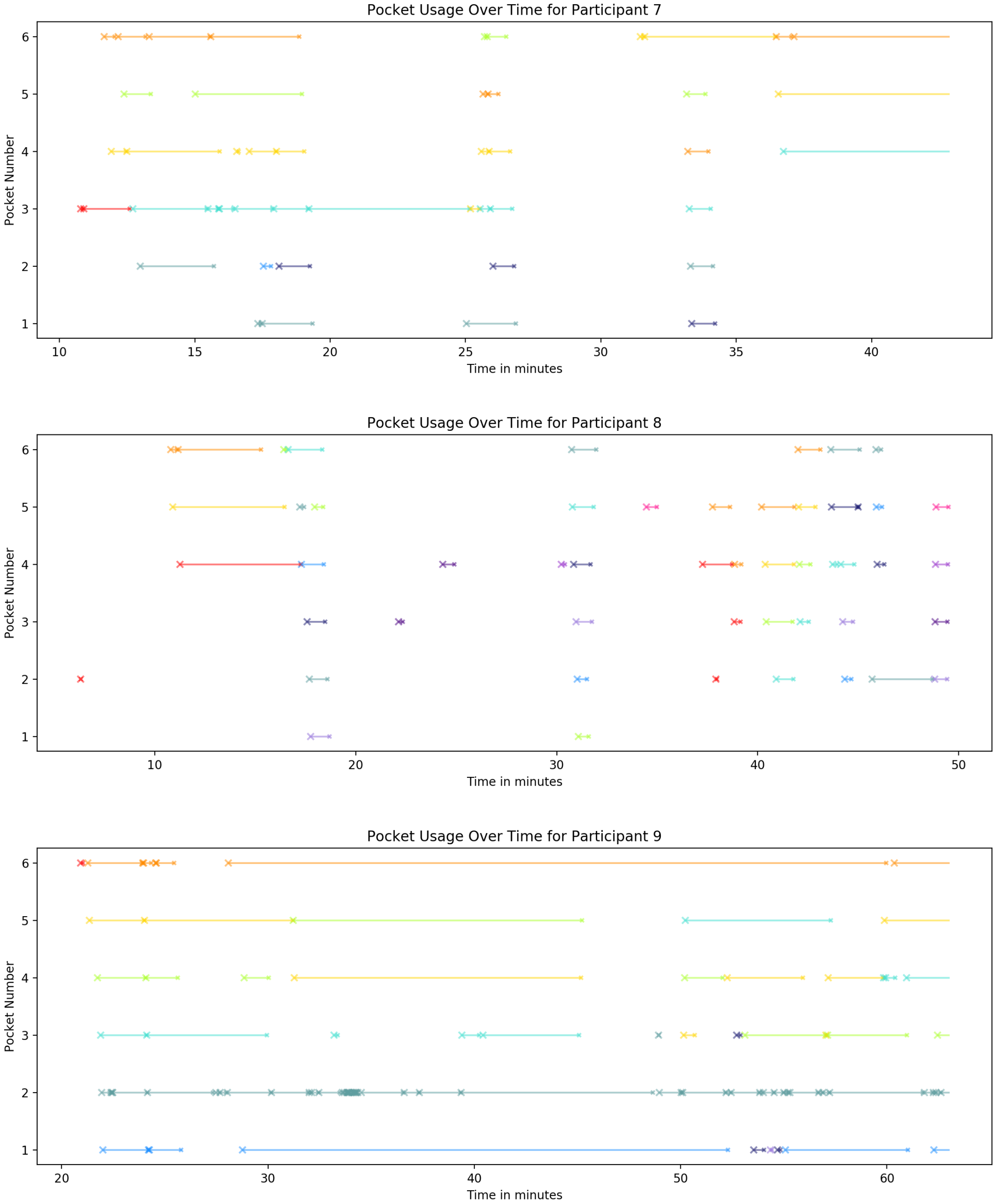


Figure 3-28: All instances of an object being taken out of or put into a Pocket, per user for participants 7, 8, and 9. Larger markers are an object being put into a Pocket, and smaller markers are an object being taken out. Markers of the same color represent the same object. Lines represent time during which the object was in the Pocket.

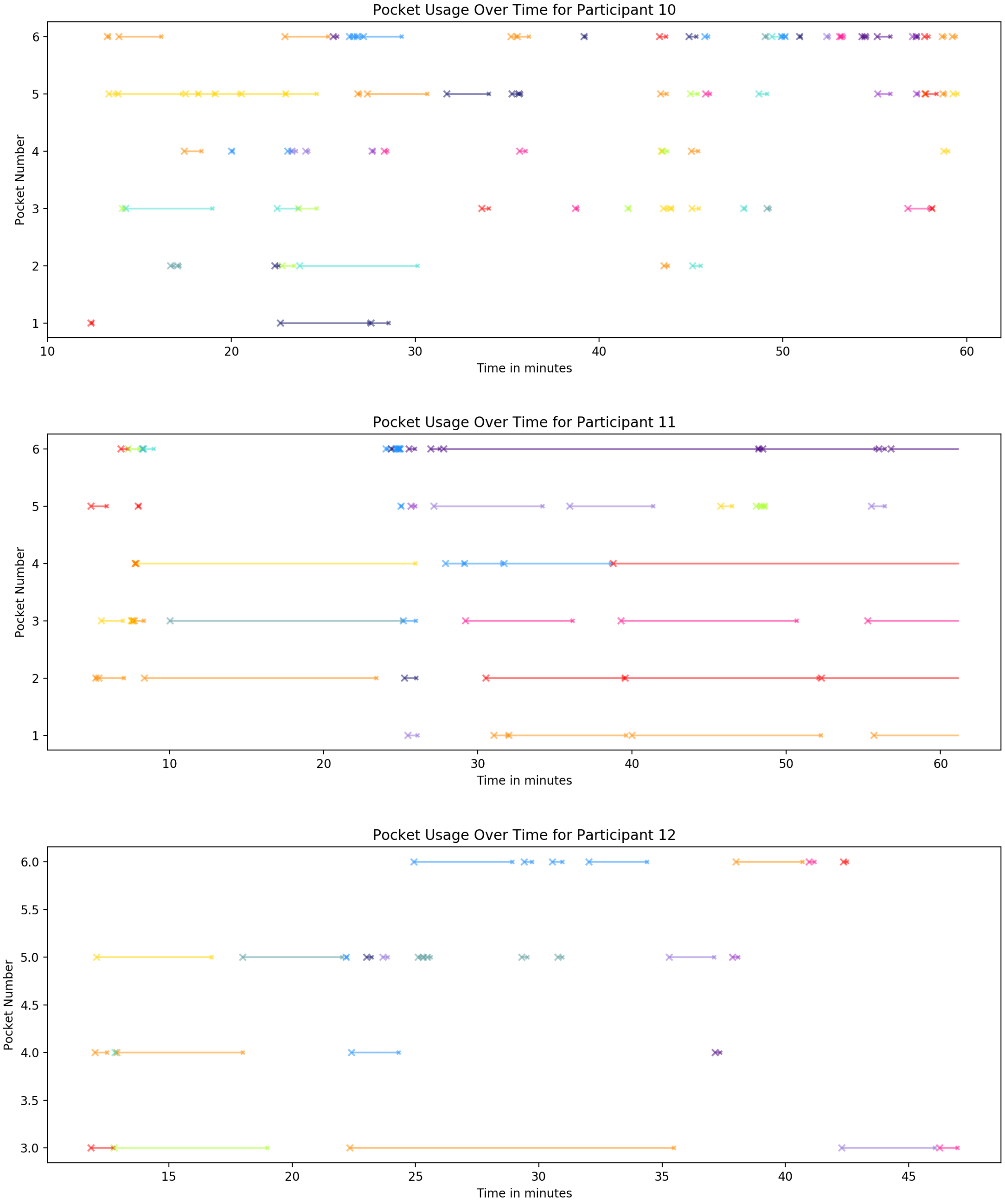


Figure 3-29: All instances of an object being taken out of or put into a Pocket, per user for participants 10, 11, and 12. Larger markers are an object being put into a Pocket, and smaller markers are an object being taken out. Markers of the same color represent the same object. Lines represent time during which the object was in the Pocket.

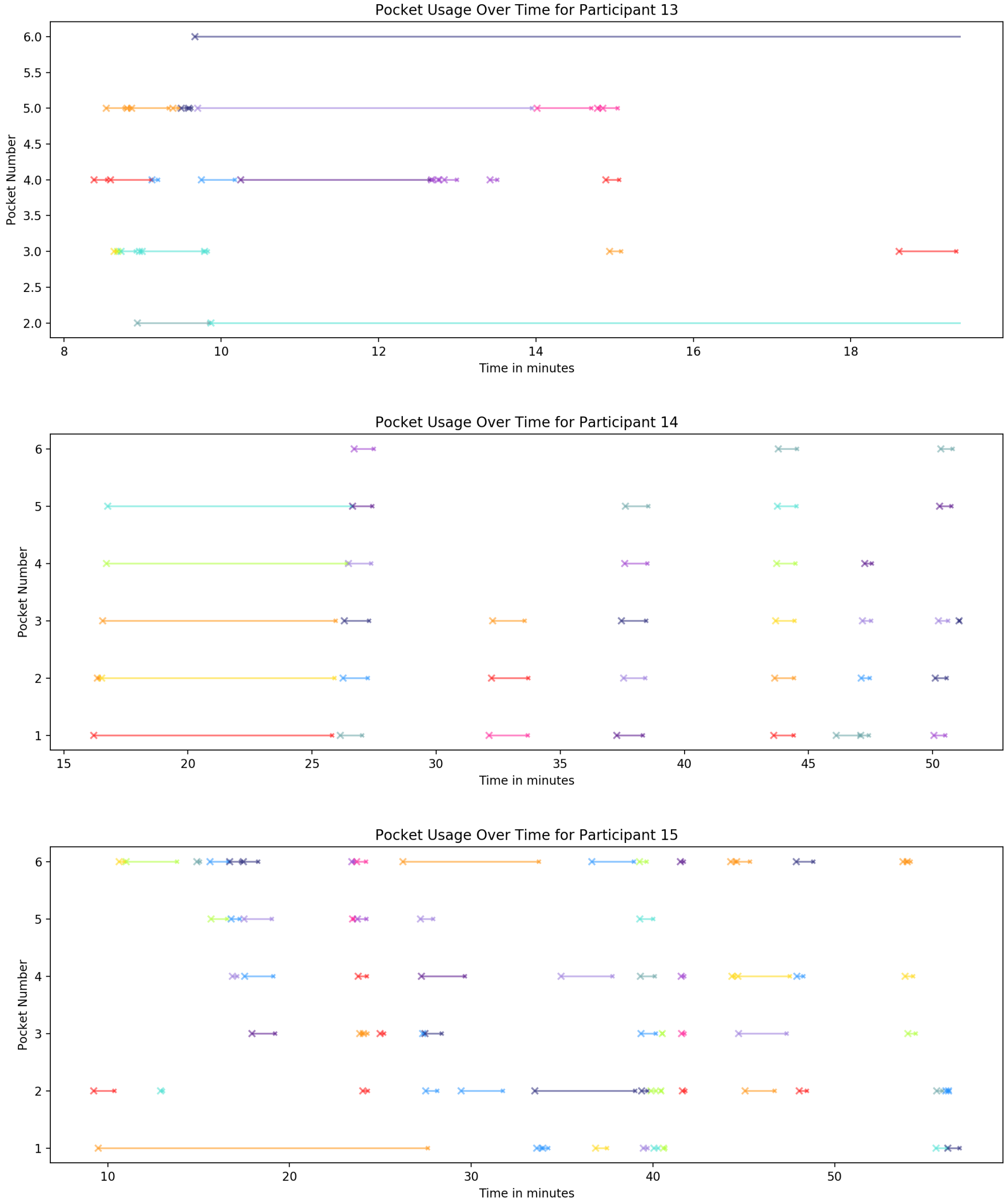


Figure 3-30: All instances of an object being taken out of or put into a Pocket, per user for participants 13, 14, and 15. Larger markers are an object being put into a Pocket, and smaller markers are an object being taken out. Markers of the same color represent the same object. Lines represent time during which the object was in the Pocket.

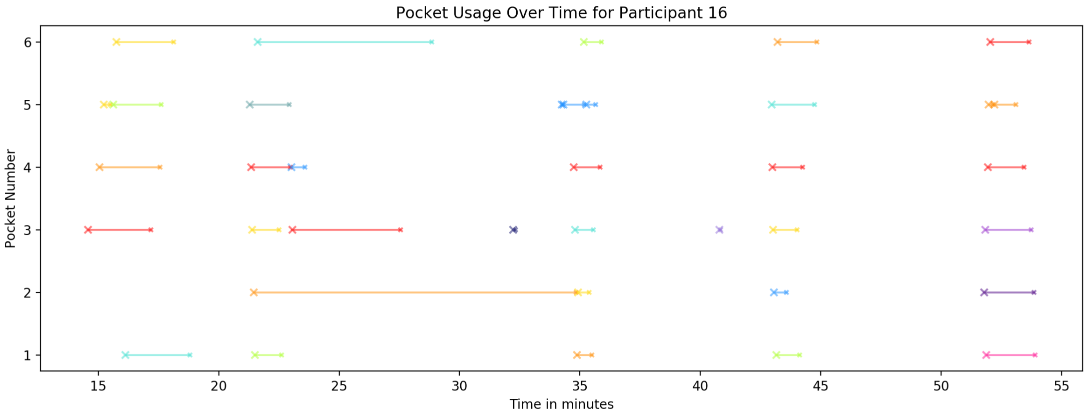


Figure 3-31: All instances of an object being taken out of or put into a Pocket, per user for participant 16. Larger markers are an object being put into a Pocket, and smaller markers are an object being taken out. Markers of the same color represent the same object. Lines represent time during which the object was in the Pocket.

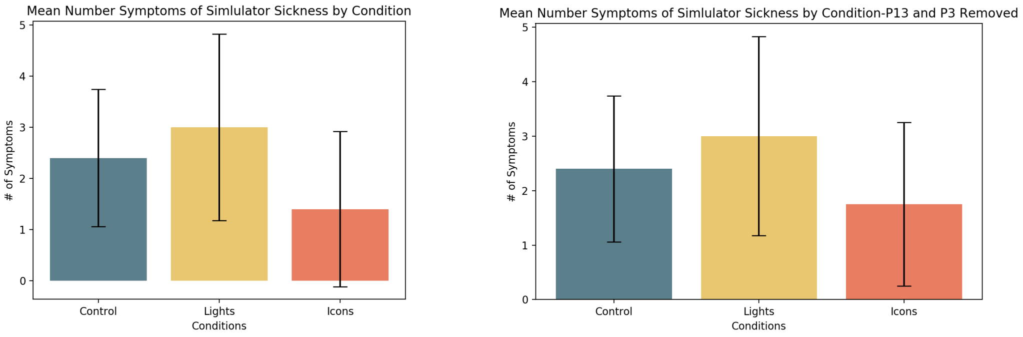


Figure 3-32: Left: Average number of simulator sickness symptoms per condition. Right: Average number of simulator sickness symptoms per condition with participants 3 and 13 removed. Error bars represent one standard deviation for each.

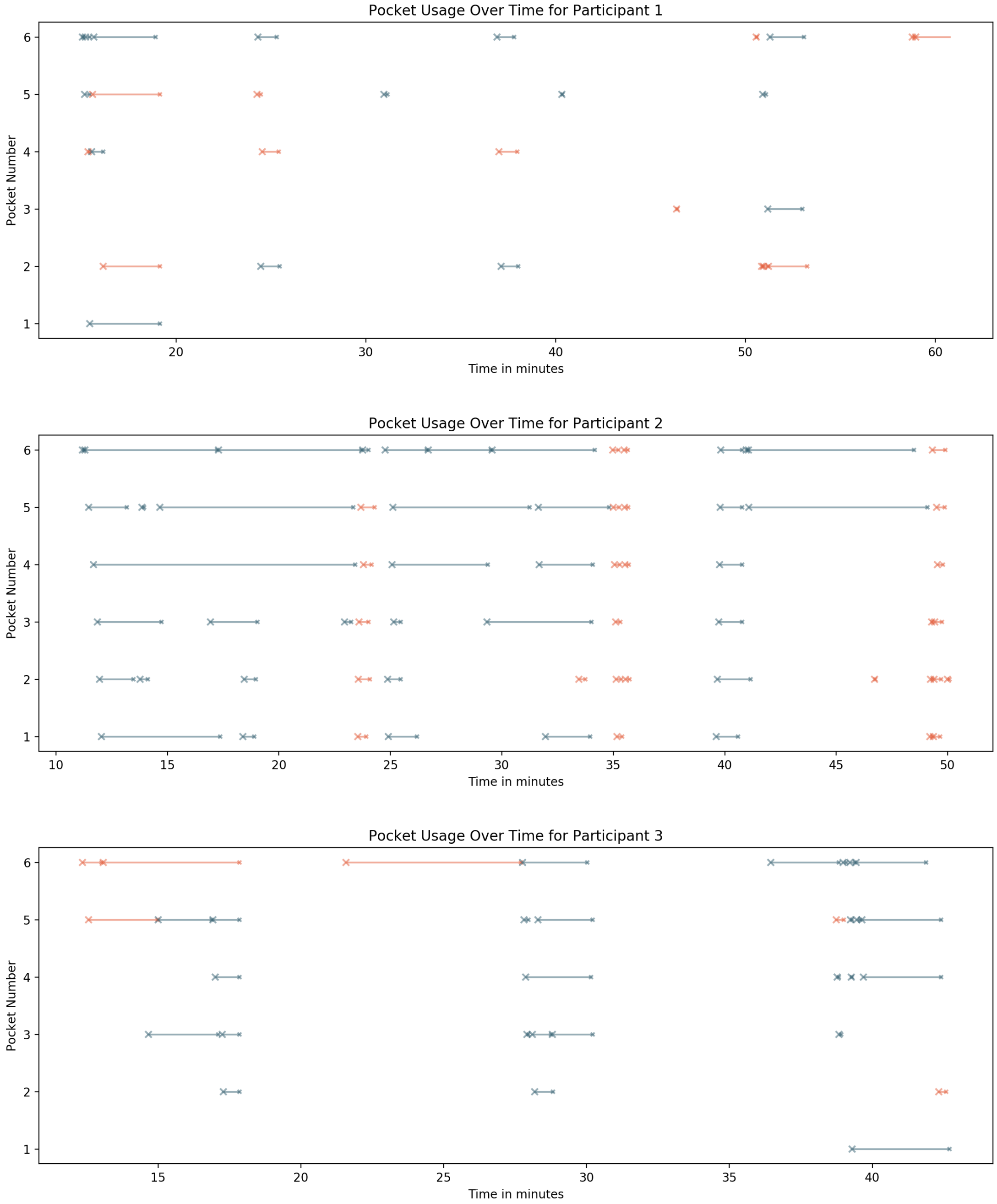


Figure 3-33: All instances of an object being taken out of or put into a Pocket, per user for participants 1, 2, and 3. Larger markers are an object being put into a Pocket, and smaller markers are an object being taken out. Blue markers represent tools of which there are only one copy such as generators and the trashcan, while orange markers represent objects of which many copies can be generated such as blobs, paintbrushes, spheres, and cubes.

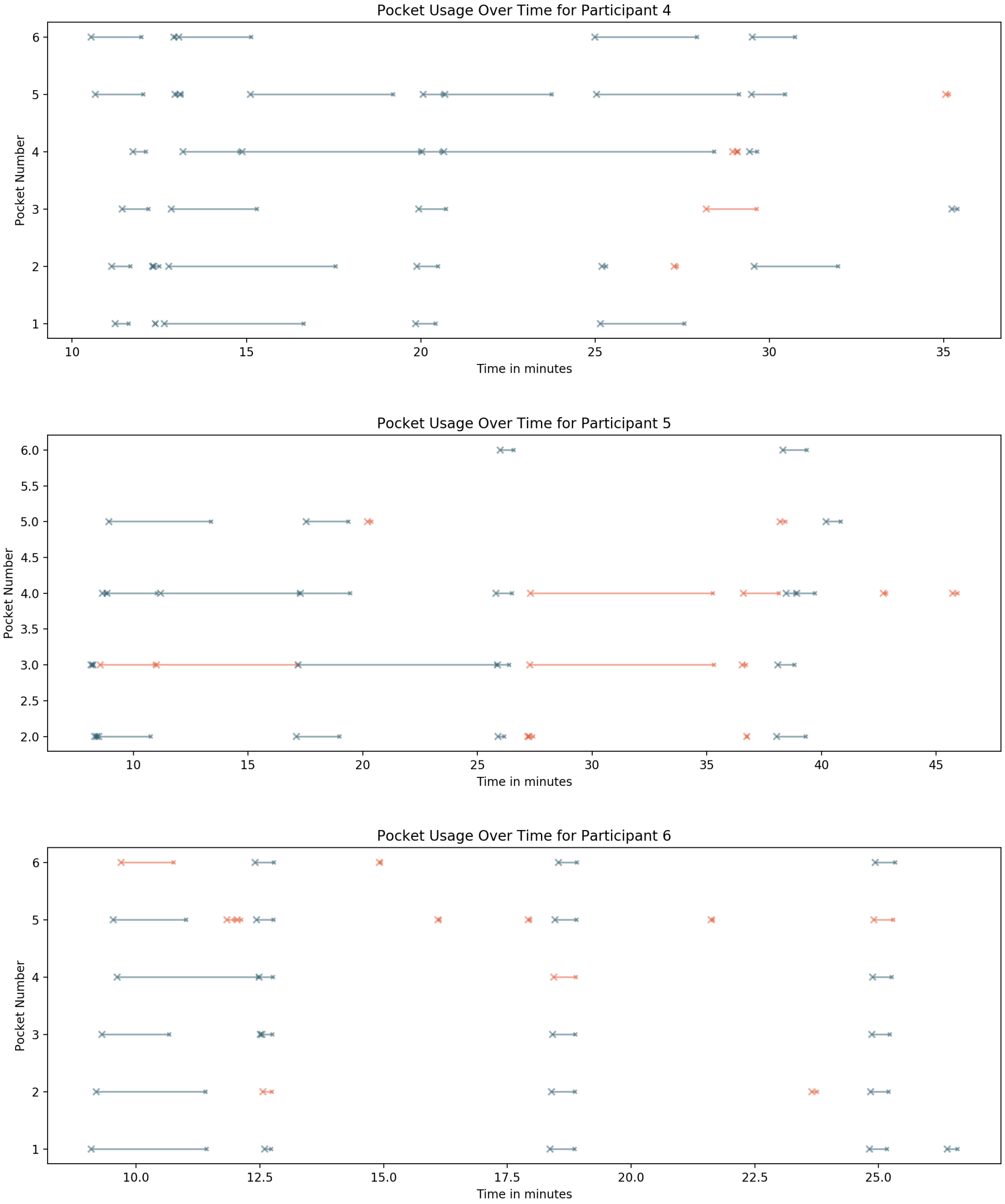
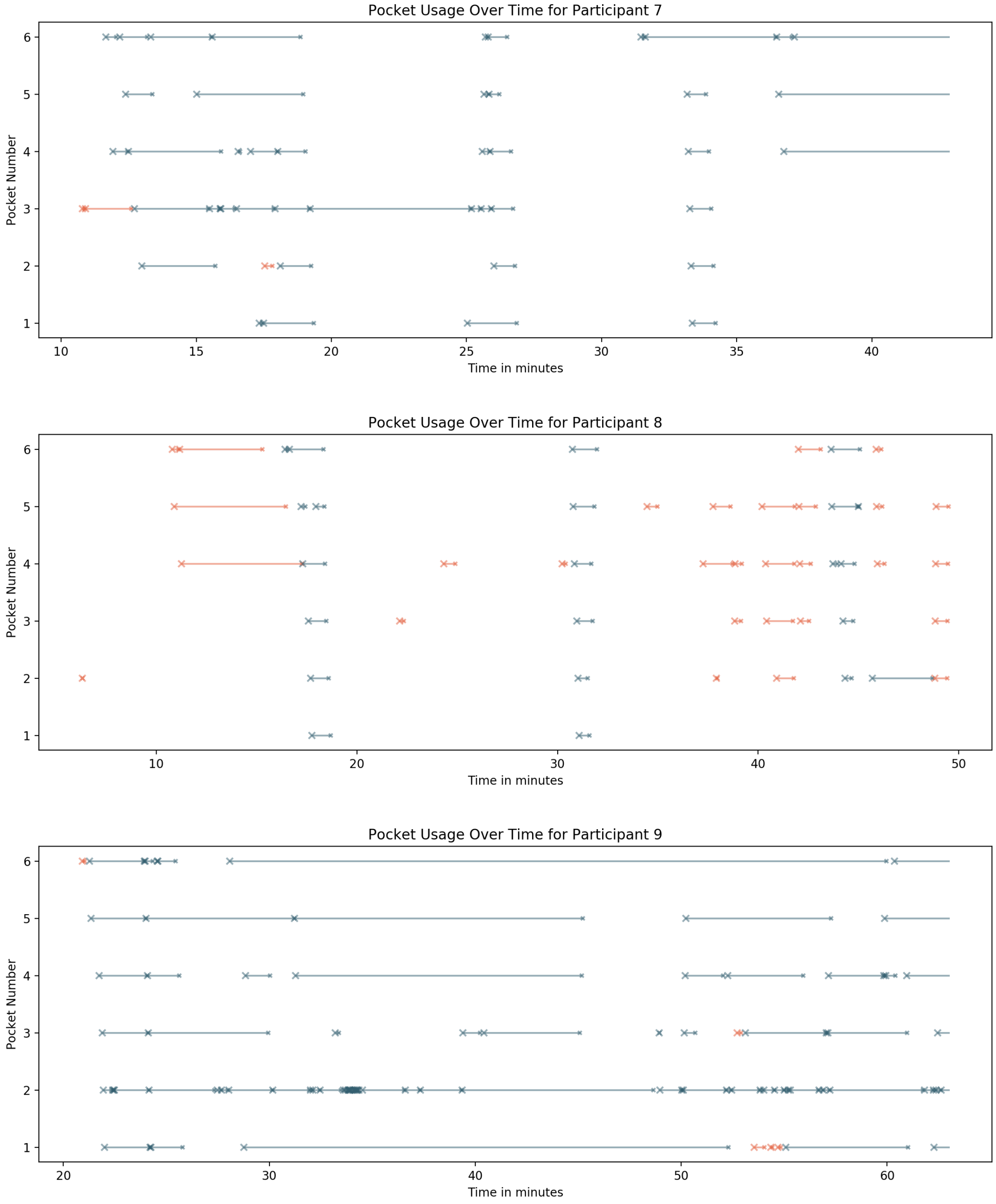


Figure 3-34: All instances of an object being taken out of or put into a Pocket, per user for participants 4, 5, and 6. Larger markers are an object being put into a Pocket, and smaller markers are an object being taken out. Blue markers represent tools of which there are only one copy such as generators and the trashcan, while orange markers represent objects of which many copies can be generated such as blobs, paintbrushes, spheres, and cubes.



Pocket Usage Over Time for Participant 7

Pocket Usage Over Time for Participant 8

Pocket Usage Over Time for Participant 9

Figure 3-35: All instances of an object being taken out of or put into a Pocket, per user for participants 7, 8, and 9. Larger markers are an object being put into a Pocket, and smaller markers are an object being taken out. Blue markers represent tools of which there are only one copy such as generators and the trashcan, while orange markers represent objects of which many copies can be generated such as blobs, paintbrushes, spheres, and cubes.

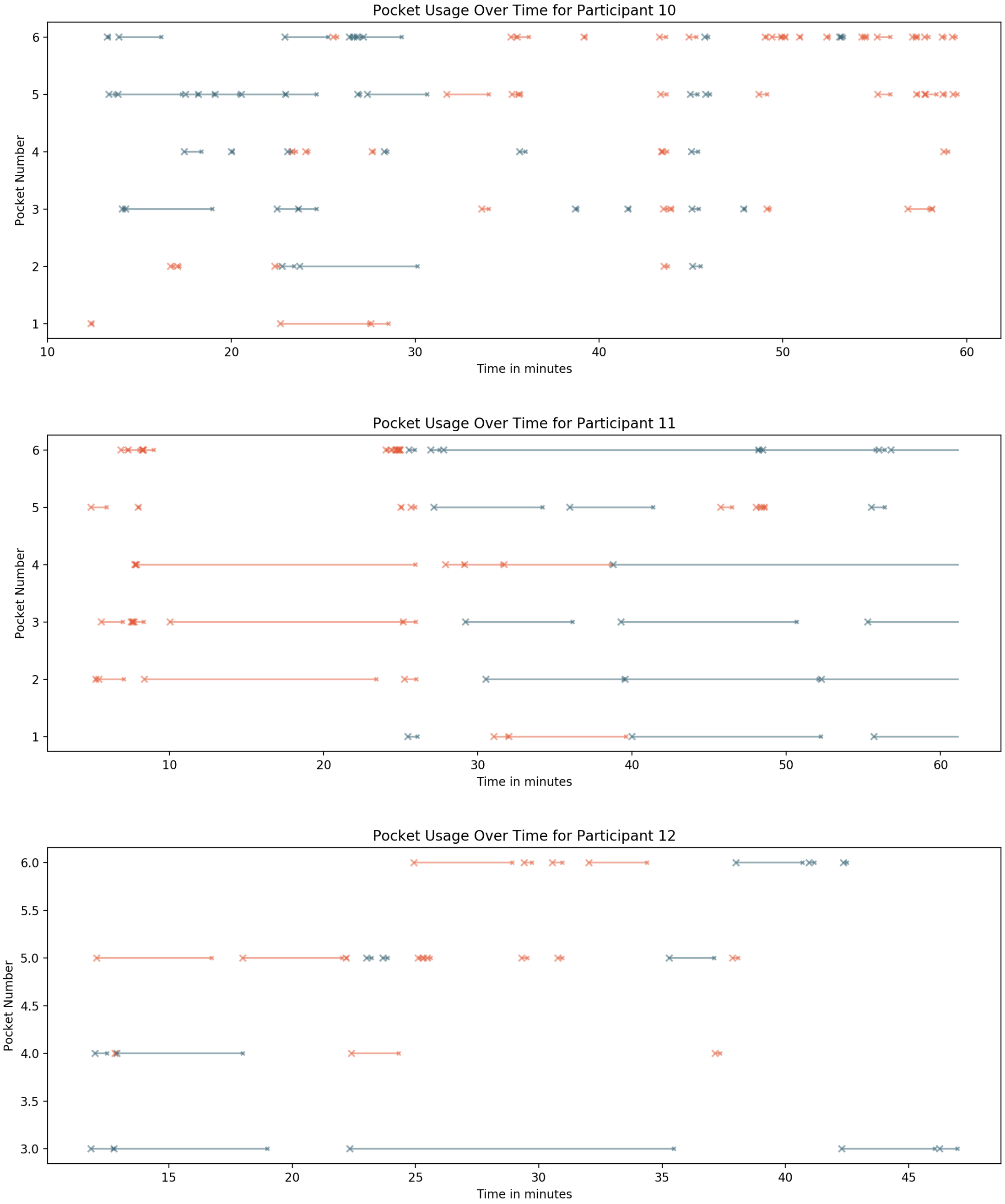


Figure 3-36: All instances of an object being taken out of or put into a Pocket, per user for participants 10, 11, and 12. Larger markers are an object being put into a Pocket, and smaller markers are an object being taken out. Blue markers represent tools of which there are only one copy such as generators and the trashcan, while orange markers represent objects of which many copies can be generated such as blobs, paintbrushes, spheres, and cubes.

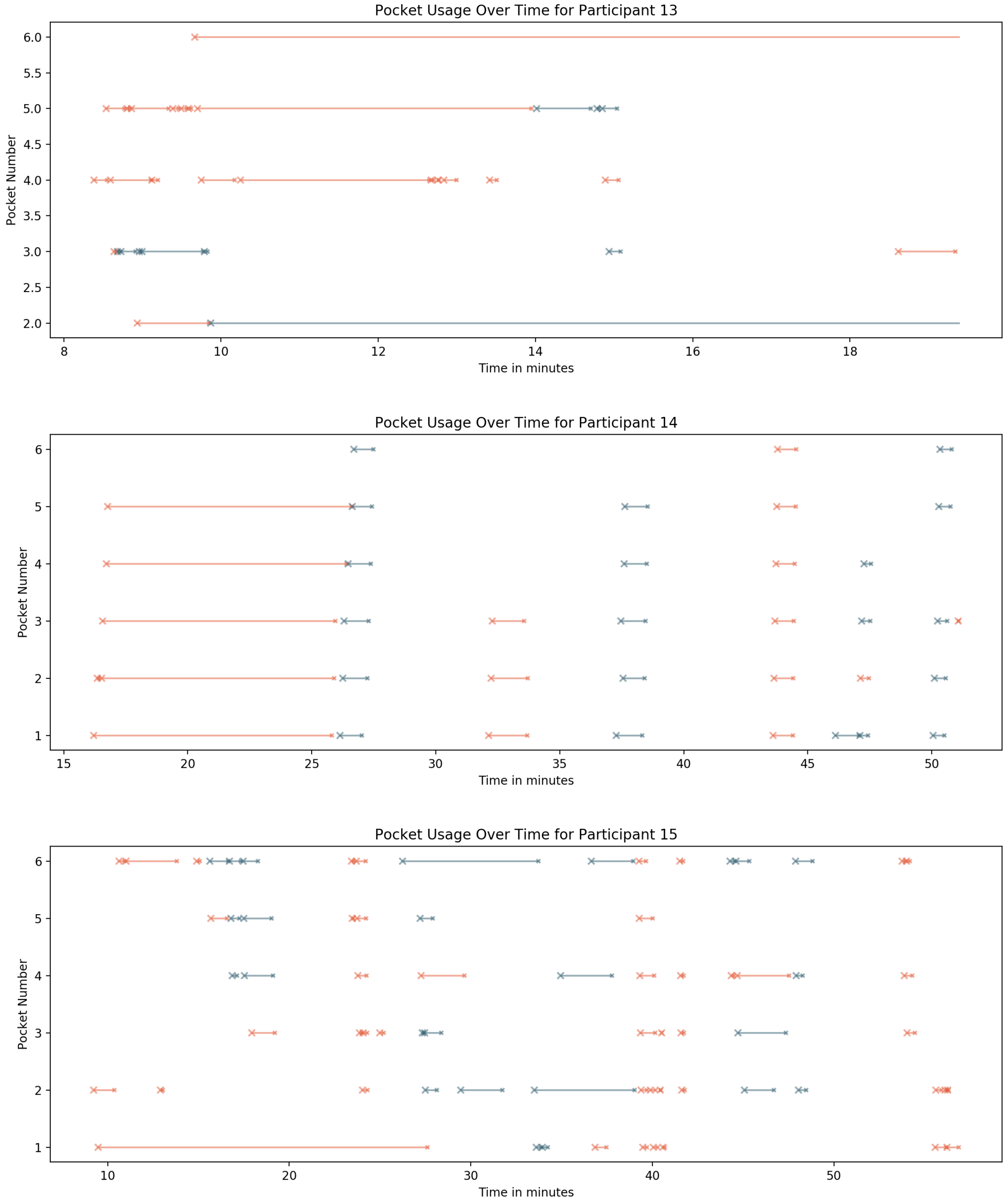


Figure 3-37: All instances of an object being taken out of or put into a Pocket, per user for participants 13, 14, and 15. Larger markers are an object being put into a Pocket, and smaller markers are an object being taken out. Blue markers represent tools of which there are only one copy such as generators and the trashcan, while orange markers represent objects of which many copies can be generated such as blobs, paintbrushes, spheres, and cubes.

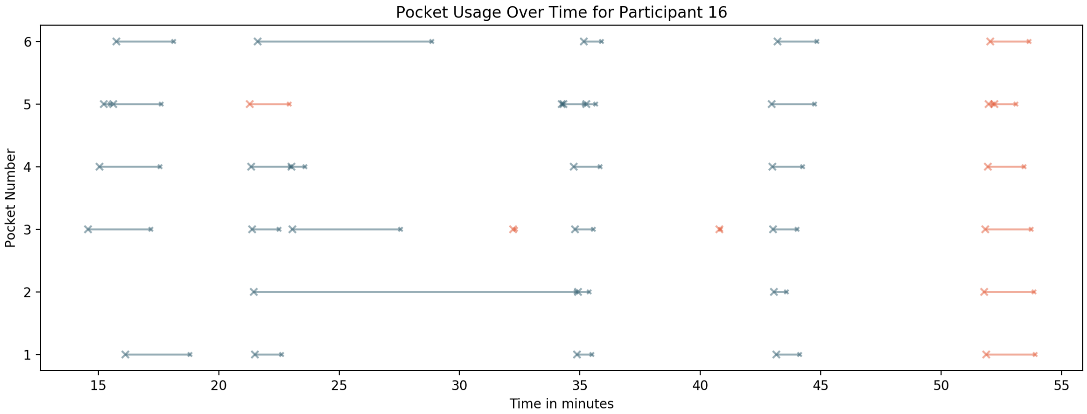


Figure 3-38: All instances of an object being taken out of or put into a Pocket, per user for participant 16. Larger markers are an object being put into a Pocket, and smaller markers are an object being taken out. Blue markers represent tools of which there are only one copy such as generators and the trashcan, while orange markers represent objects of which many copies can be generated such as blobs, paintbrushes, spheres, and cubes.

confidently, “Oh I see what I can do here. I can put this in a Pocket, and then I can go here, and then I can take it out.” Participant 14 considered Pockets to be, "great for moving tools virtually around the space." Participant 15 said that, "the main problem is like to carry things from somewhere to somewhere else," and that, "in general the idea of carrying things in Pockets is great." Several participants therefore didn’t see forgetting what was in each of the 6 Pockets as a goal. Participant 5 (Lights condition) "was more focused on creating the sculpture that I didn’t even consider memorizing what was in each pocket."

This means that dumping out the contents of Pockets suddenly, which I had taken to be a marker exclusively of forgetting what was in them, was sometimes instead a marker of a suddenly shifting need in terms of what to carry. When participant 6 walked to a sculpture, they started copying it by saying "Let’s just turn our pockets inside out, because I know I’m going to need all of this stuff." Participant 11 (Lights condition) spent "a little time orienting things so that I can see them, putting them in a close proximity to me,” and then would, "carry the tools into the next room with me," before setting up the "next workstation".

Many participants would make all of the components near the sculpture to copy, and would then put them all in Pockets in order to carry them to the other pedestal. This can be seen in figures 3-33-3-38.

This suggests something interesting about Pockets, and their role in virtual reality as a whole. Menus are, fundamentally, a computing concept. In physical reality, we don't interact with the world via a menu. We interact with objects, tools, and the world in a more direct and physical way. Pockets ultimately aren't a menu system so much as they're a means of supporting an alternative interaction paradigm to menus–one based on 3D tools. Participant comments suggest that they treated it as such.

Future Work

Perhaps the most essential and immediate future work for Pockets, and this study in particular, is exploring Hypotheses 1, 2, and 4. While the necessary data was collected during this study, by the writing of this document, the necessary data analysis has not been done. This data can be found at the code link listed in appendix A. I intend to complete this data analysis by November of 2021, for submission to IEEVR as a conference paper. There are an enormous number of minor variations on the basic concept of Pockets to be explored, such as the number, design, and placement of buttons. There are also several interesting open questions regarding how Pockets compare to traditional menus of various types for various tasks.

Other future work related to Pockets as a whole is discussed in section 4.2.

3.2.5 Conclusion

This chapter explores the second tool for interacting with virtual environments introduced by this thesis. It proposes an alternative for traditional flat panel menu interactions which is more fundamentally 3D and aligned with our experience of physical reality, and evaluates this alternative using a hardware prototype and two user studies. The first of these studies is preliminary, and the second explores 3 different means of representing Pockets and their contents in the virtual environment. While more data analysis is necessary to draw firm conclusions from this study, results thus far suggest that visually representing both the Pockets and their contents in the virtual environment is preferable to representing only the Pockets and if they are full, or not representing the Pockets at all.

Perhaps the most interesting comment from a participant was from experiment 2, participant 11, who said:

> "It's a combination of my computer skills and my real world skills. In the way of the windows mouse desktop, which I'm old enough to have lived through the birth of, but like a leap higher."

This is exactly what I hope Pockets can be– an organizing UI concept which can undergird a leap higher, and a change in how we think about VR interaction on the scale of the introduction of WIMP interfaces in 2D computing.

Chapter 4

Unimplemented Designs for Future Work

4.1 Why is the word "Integrated" in the title of this thesis?

It is notable that the word "integrated" is a in the title of this thesis which presents two distinctly non-integrated interaction techniques. While these techniques were never implemented together, they are designed to fit together into one modular system built on Pockets. Pockets allows users to store and carry 3D tools. The Bird is an instance of one such 3D tool. It is a selection and manipulation technique, but is by no means meant to be the only selection and manipulation technique. It would work very poorly in situations which require direct manipulation, and is slower than a ray-cast when interacting with a virtual environment full of panel-based menus.

To return to the design principles discussed in 1.4, and in particular the discussion of modularity in 1.4.1, designing systems from their earliest phases with loose interfaces which can accommodate a variety of potential inputs allows lead users to better adapt systems to fit different needs. For example, a user with a repetitive strain injury may find it uncomfortable to use the Bird for extended periods of time. The modularity of Pockets could allow switching between a less convenient, but less

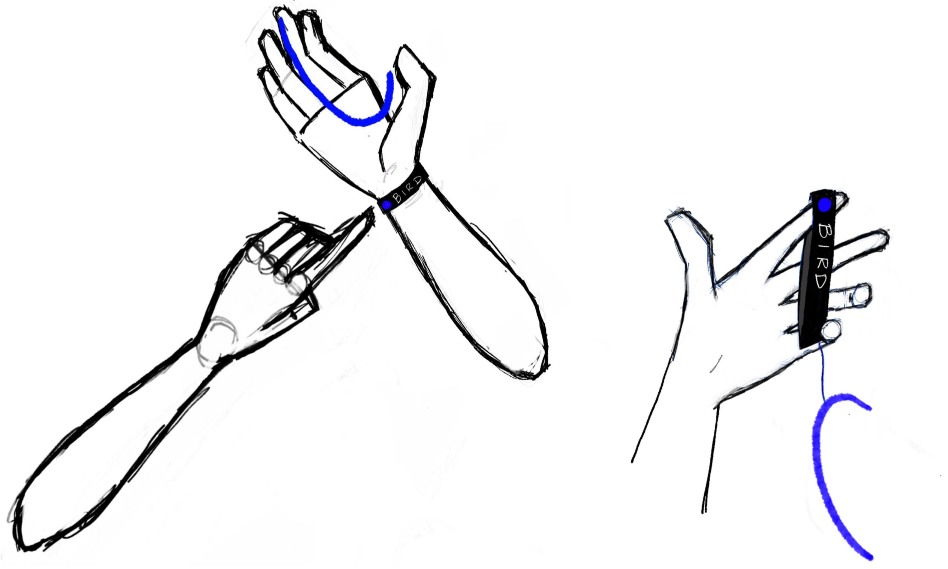


Figure 4-1: Taking off the Bird, and the Bird as a 3D tool while not engaged. This version includes an arc to visualize the overall openness of the hand, and therefore distance of the cursor.

uncomfortable point cursor and the Bird.

The most obvious means I could imagine by which to facilitate shifting between the Bird and other selection and manipulation techniques would be to have a clear visual handle on the dominant hand, or whichever hand is currently using the Bird, which could be grabbed using direct manipulation with the other hand and pulled off in a specific direction, much like taking off latex gloves. Pockets and the Bird are designed to be integrated into one and the same system.

In section 4.3, I cover a number of additional techniques which integrate well with the Bird and Pockets, but were not implemented due to the limited scope of a masters thesis.

4.2 Essential Future Work on Pockets and Bird

The work in this thesis was severely limited by the COVID-19 Pandemic and related restrictions. In particular, access to in-person user studies, machine shops, and tracking equipment was severely limited, making it difficult to implement these techniques in ways which allowed them to be evaluated effectively. Were this work to have been conducted, or to be continued, during less unprecedented times, the following would be essential continuations.

Bird with Motion Capture

As discussed in 2.2.4 and 2.2.6, both the data glove and the hand tracking native to the Oculus Quest caused considerable difficulty for participants. Ideally, the Bird would be re-implemented with non-consumer motion capture such as an Optitrack system, which is more stable. This would make it possible to more fully isolate the usability of the technique from the usability of the hardware which supports one particular implementation.

Pockets with Motion Capture

The conditions for the second experiment with Pockets(3.2.4) were also severely limited by the available tools. In the visible conditions, Pockets were visualized as a heads up display. While this was partly due to the fact that this is an interesting implementation– it's easy to implement with current hardware, and unclear if it causes greater simulator sickness– it was also chosen partly because it's what was possible.

Pockets, as a floppy wearable device, was extremely difficult to track with consumer equipment. While it was possible to track the overall location of the tool belt using a Vive, it was impossible to track the specific location of each of the six Pockets with any useful level of accuracy. As such, it was impossible to co-locate the visual representation of Pockets with their physical counterparts in a manner which was worthwhile to evaluate. It would be extremely interesting to use a motion capture system (again, likely an Optitrack) to simulate reasonable future advances in optical tracking on consumer equipment, track the location of each of the buttons on the tool belt, and co-locate visualizations of Pockets and their contents inside of the headset with their physical counterparts.

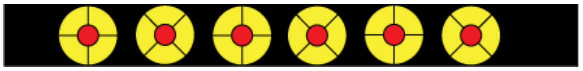


Figure 4-2: Button covers which recess buttons, protecting them from being accidentally triggered by elbows, and make them larger and easier to find. Alternating patterns of raised ridges, or other textures may help users find the center of the button, and differentiate them from one another tactilely.

Pockets Redesign

The majority of participants in both studies accidentally triggered Pockets multiple times during the study. This seems like a completely avoidable situation with minor modifications to the buttons. In particular, covers over the buttons which slightly recess them would both make them larger and easier to find, and keep them from being pressed accidentally. This idea is vaguely bio-inspired, based on the way in which flowers manage to both attract pollinators with large displays of petals, and guide them to the much smaller relevant organs of the plant. Participants also suggested varying the textures of Pockets, to make it easier to tell them apart. Getting this right may involve some experimentation with materials and designs which would be made much easier by access to a machine shop or fabrication lab.

4.3 Other Techniques

4.3.1 Locomotion Between Spaces– Portals

Scene changes in virtual reality currently are usually either abrupt, hard cuts which are disorienting, or a fade to black/a loading screen followed by a fade to the new space. These types of scene changes also don't provide an easy way to preview the space that you're going to before you get there.

Portals are a tool for moving between spaces, and an example of a 3D tool which could be stored in a Pocket. Portals are moveable circular doorways which connect two locations. They can be grabbed and handled with the hands, stretched larger, compressed smaller, and passed over the body like a hula hoop.

An interesting exploratory study of Portals could have participants navigate a

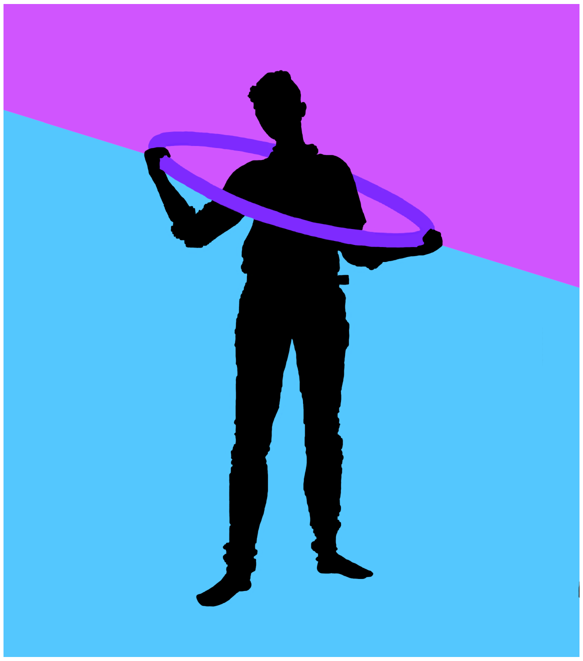


Figure 4-3: There are two different virtual environments on each side of the Portal. Moving through the Portal moves the user from one environment to the other.

system of rooms which are connected by Portals, a different but similar system of rooms which are connected by simple doorways which can be walked through, and a third system of rooms which are not connected, but which can be navigated via a menu and fade-in/ fade-out scene changes. This study could focus on determining if Portals cause significantly more simulator sickness than traditional navigation techniques, and how participants form a mental model of the layout of spaces which are connected by this non-traditional technique.

Portals also open interesting technical questions about rendering efficiency and clipping algorithms. They necessarily require rendering multiple virtual environments simultaneously, and destroy the ability to use discrete scenes to reduce memory usage. How do we render windows into virtual environments in the most efficient manner possible, and smooth the transition between virtual environments effectively, à la Netflix's distributing content in servers based on user viewership habits?

4.3.2 Locomotion Within a Space– EEG

Unlike locomotion between spaces, locomotion within a space is a strong contender for the most studied, most important, and most challenging interaction design question for virtual reality today. An essential promise of virtual reality is the ability to experience locations without being there. If those locations are going to be larger than the physical space in which they are experienced, a metaphor for locomotion is necessary.

It is in particular difficult to allow a user to move through virtual space without causing simulator sickness or disrupting immersion. Teleportation is a commonly used locomotion metaphor. In most implementations, the user points a ray-cast of some type at the ground, confirms the selection, and is automatically moved to the location pointed to. However, the experience of teleporting through a space tends to be disruptive to immersion, and is experientially different from walking through a space [63]. Other forms of locomotion through a space which qualitatively feel more like walking, such as walking in place or various forms of flying (pushing a button or a joystick to move in a specific direction) tend to cause severe simulator sickness [11]. Various forms of redirected walking have been proposed in recent years, but all usually require an amount of space which is impractical for day to day use [74] [75] [17] [54] [35]. [71] explored redirected walking and reorientation, in addition to means of optimizing the use of a user's physical space in the creation of virtual environments. This is by no means an exhaustive overview of work on locomotion within a space in virtual reality.

An interesting emerging means of locomotion within a space in VR which maintains high immersion without causing simulator sickness is to use brain-computer interfaces to control walking. [43] explored BCI input for a non-VR video game. They integrated BCI into a modified version of Doom 3. With only a minimal training phase during the game itself, participants were able to select either a flashlight or weapon only by visualizing the item with 72% accuracy for the most effective condition. BCI and machine learning for binary choices in video games has already been used for locomotion in VR. In a very preliminary but promising study, [4] allowed users, after a training phase, to move forwards or backwards in a CAVE environment.

This technology is still in very early phases of development, and is by no means ready to become the primary way in which users move around in virtual environments yet. However, as both consumer-grade EEG technology and machine learning improve, improvement from binary choices with relatively low detection accuracy to ternary choices with somewhat higher accuracy seems likely. This would allow users to move forward, turn left, or turn right using only their minds, and opens fascinating near future possibilities of walking in VR by imagining walking in VR.

4.3.3 Text Entry– Roll and Pinch Keyboard

Text entry in the absence of a physical keyboard is a well-studied problem. In 2002, [15] compared "the Pinch Keyboard (a typing emulation technique using pinch gloves), a one-hand chord keyboard, a soft keyboard using a pen tablet, and speech" and found that "no single technique exhibited high levels of performance, usability and user satisfaction."

PinchType is a means of text entry for virtual environments in which users touch their thumb to their pointer finger in order to type letters[27]. An individual touch could correspond to any of the keys which that finger could be used to press on a keyboard. For example, touching the thumb to the pointer finger on the left hand could correspond to "R", "T", "F", "G" or "B". A language model then determines the most likely word based on those inputs. The list of most likely words appears to the user's right, and it is possible to shift to other words in the language model. What the paper does not clarify is how to type words which are not included in the language model, and what words are and or not included in the language model. If broadly used, this has interesting implications related to linguistic evolution. For example, would AAVE words be included in the language model? Would names, and, if so, which ones?

The Roll and Pinch or RAPMenu is a radial menu system which allows for the selection of up to 12 unique items per hand using the combination of touching the thumb to each of the four fingers in up to 3 workspaces based on the rotation of the hand[65]. While the RAPMenu was explored as a tool for menus, with specific options like "Music" or "YouTube" assigned to specific menu locations, I think it's a promising method for text entry. Assuming that hand tracking improvements since the initial publication of the paper in 2008 would allow for the detection of 4 different comfortably accessed and recognized rotational positions per hand, this would allow for 16 unique items per hand, or 32 possible options without submenus, which seems to be enough to implement a keyboard. If chords are used, and each combination of a left and right handed pinch corresponds to a separate "key", 24 possible combinations are possible per workspace, which would allow for 94 possible inputs without submenus.

Another potential solution for text entry in VR, which is less heavily based on desktop computing, would be to look for text-entry systems without keyboards which already exist. Finger-spelling in American Sign Language requires only one gesture per character, and has been optimized for speed and comfort via being used as a language. Finger spelling allows for the description of all 26 characters of the alphabet with one hand, which leaves the non-dominant hand free for control keys. Several finger-spelling classifiers also already exist [73]. ASL finger-spelling as a text input for VR technique seems, from a cursory review of the literature, to be an unexplored space.

A comparative study between various key layouts of the RAPKeyboard, ASL Finger-Spelling, and PokeType, which is the current standard for rapid text input in VR, seems like an extremely rich area of future work. Both the RAPKeyboard and Finger-Spelling are also the variety of technique which could easily be encapsulated into a 3D tool, shifted between, and stored in a Pocket[26].

4.3.4 Pockets Extension- Bins

Bins are a hypothetical means of expanding the number of objects which can be stored in a Pocket. A Bin is a virtual object which contains other objects. Objects can be placed into the Bin by physically intersecting their collider with that of the Bin in a virtual environment. Objects inside of a Bin are not active while in the Bin–

they are not rendered, and do not participate in physics in any way. A Bin can be emptied by turning it upside down, which should scatter all of the contents of a Bin within a reasonable area– large enough to see everything, but small enough to easily return objects to the Bin.

Interesting aspects of Bins to explore in future experimental work include limiting their capacity, the density at which they distribute objects which have been dumped out, and means of labeling them or otherwise representing the objects inside of them.

4.3.5 Bird Extension- Skitter

Skitter is a means of including pseudohaptics in Bird. Skitter was inspired by the difficulty with placing objects flat on the floor, rather than partially through the floor, using the Bird. Skitter causes the Bird to stop moving forward when it collides with a surface that should be solid, and slide along it rather than going through it. When it comes off of that surface, it linearly interpolates back to where is should be. It would be interesting to compare this to an unmodified version of the Bird, and to explore different speeds of snapping back to the "right" location after sliding off of a surface.

4.3.6 Shared Spaces with Desktop-Based Participants- Puppet Avatars

HMD ownership is not as common as laptop ownership. Therefore, in current virtual immersive environments for both HMD-based and browser-based users, many, if not most users tend to be browser-based. Until a potential future in which presence in these mixed spaces is HMD-based by default, it is essential to ensure that mixed browser and HMD based virtual immersive environments are an engaging experience for all of their users. HMD ownership is limited by the current cost of systems, as well as physiological constraints– people prone to motion sickness are unable to use them. As part of a commitment to modularity and accessibility, I therefore explored desktop-based experiences as part of my work on this thesis.

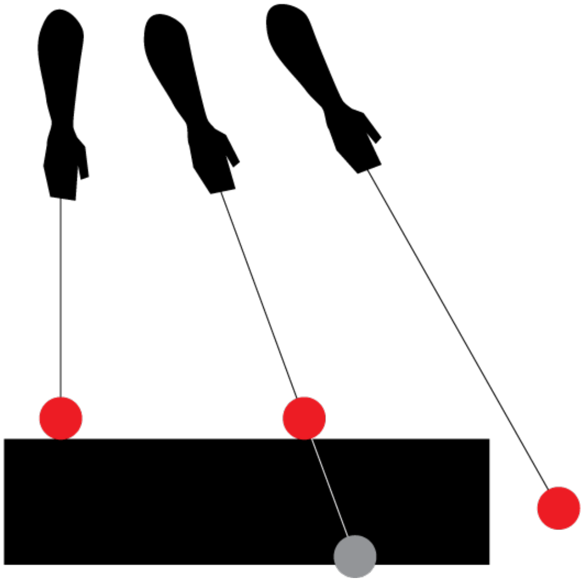


Figure 4-4: Skitter allows the Bird to slide across "solid" surfaces, rather than phasing through them. The gray sphere represents the position the Bird would be at based on the hand position, and the red sphere represents it's actual position as it is moved across a surface.

With the introduction of WebGL-based virtual reality and applications like Mozilla Hubs, virtual environments which can be accessed either via head-mounted display (HMD) or via the browser have become increasingly popular event spaces [2]. For HMD-based users, experiences such as Mozilla Hubs in particular can be very immersive, and avatars can be very expressive. HMD-based users can control their viewport and the movement of their avatar's head using their own physical head. They also have the ability, if they have controllers, to point and gesture in a manner which is very expressive using their physical hands. Browser-based users lack this expressiveness. They generally have no hands or ability to point. The control scheme for browser-based users also isn't particularly embodied. The viewport and avatar head are moved by clicking and dragging with the hands, rather than by moving the head.

For both browser-based and HMD-based users, many virtual immersive environments use avatar idle movements, especially of the eyes, which are not driven by the user's actual body[2]. Other applications avoid showing eye movement by hiding and avatar's eyes, both for HMD-based and browser-based users. Arthur does this by adding opaque black sunglasses to all avatars[1]. While idle movements can help make an avatar seem more lifelike, they also provide false social cues which may not match with the actual body language of a user. This can be especially troublesome for browser-based users, whose avatars continue to blink attentively when the users they represent may be in a different tab, or away from their keyboards.

While browser-based users in virtual immersive environments are not using a device which can track headpose and handpose, most personal laptops have many affordances which are not utilized by existing virtual immersive environments. Generally, a laptop computer has a 2DOF trackpad and a camera. We propose an alternate control scheme which utilizes the trackpad and camera to drive attributes of a simple avatar.

I call them puppet avatars because, while they cannot create the same level of embodiment as an HMD, I aspire to create a type and level of embodiment akin to that of manipulating a puppet.

Puppet avatars are very abstract. This is partially to reduce the strain on bandwidth involved in transmitting video data, and the computation involved in animating more complex avatars and interpreting facial expressions. It also avoids the deeply unsettling glitches which can occur when face or especially limb tracking mapped to an avatar on a computationally limited device fails. This is just one possible configuration in what we hope to be an ongoing discussion of browser-based avatar embodiment. Potential expansions on the mappings shown in 4-5 include facial rigging mapped to the user's own facial movements, especially of the mouth, or mapping the avatar's limbs to the user's limbs via skeleton tracking, thereby allowing them to gesture with their physical arms [37].

For reasons related to both privacy and accessibility, each feature of this control scheme is conceptually modular, and able to be turned off without affecting other features. For example, a user who does not want to turn their camera on or does not have a functional camera would not be able to move their avatar's head by moving their own, but this should not prevent them from pointing using the trackpad.

While Puppet Avatars were not implemented in the course of this work, the idea was presented as part of the CHI2021 SocialVR workshop.

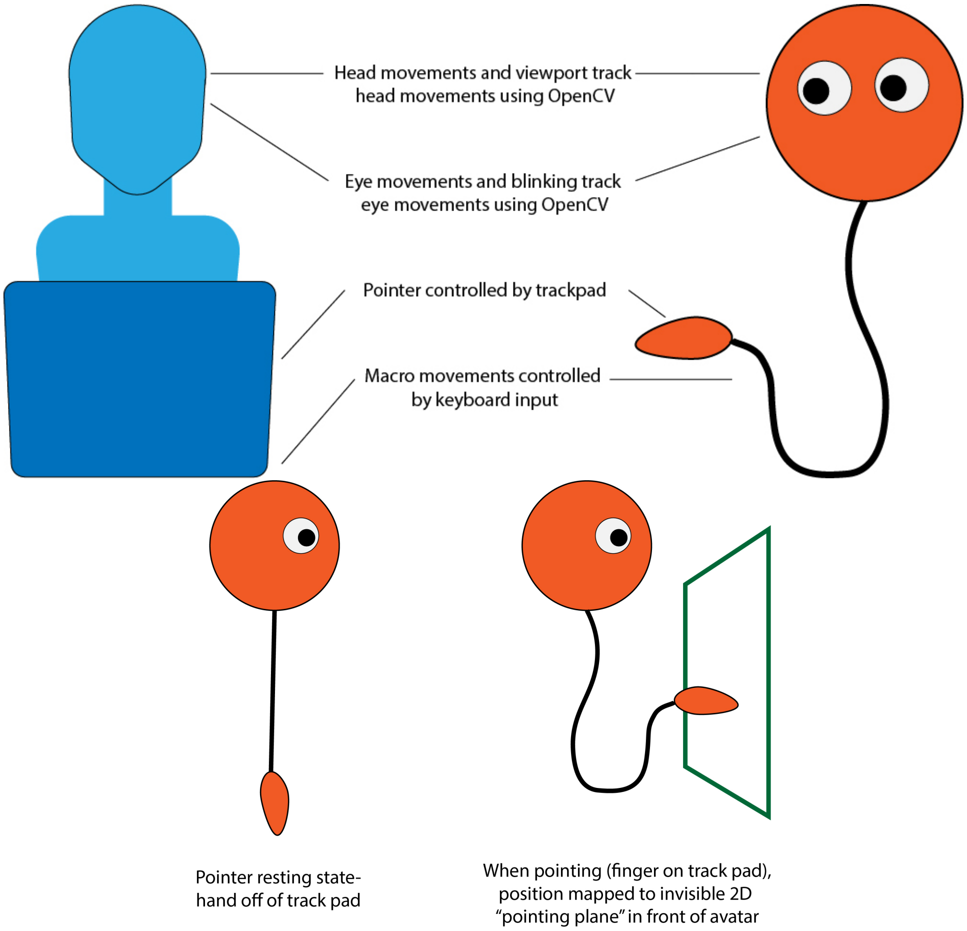

Head movements and viewport track head movements using OpenCV
Eye movements and blinking track eye movements using OpenCV
Pointer controlled by trackpad
Macro movements controlled by keyboard input
Pointer resting state- hand off of track pad
When pointing (finger on track pad), position mapped to invisible 2D "pointing plane" in front of avatar

Figure 4-5: A means of controlling an avatar in a virtual environment using a laptop computer which is more embodied than using the mouse to move the viewport. Using the webcam, the head and eye movements of the user control those of the avatar, and make adjustments to the viewport, allowing the user to look around in a limited range using their physical head. These simple avatars also control a hand-like "pointer", controlled using a trackpad, which allows for some ability to point and gesture.

Chapter 5

Conclusion

5.1 Summary of Contributions

In this thesis, I have presented the design and implementations of two interaction techniques for virtual immersive environments. The first, Bird, a tool for selection and manipulation, was evaluated with an in-person study which compares it to similar techniques from the literature, and a remote study which compares variations on disambiguation of which item to select and which finger to select with. The second, Pockets, a tool for carrying 3D tools in virtual environments, was evaluated in a remote pilot study, and an in-person study which explored variations on visualizing the contents of Pockets. This thesis additionally presents a set of 3D tools which formed a simple 3D modeling program, as well as techniques for locomotion within a virtual environment, locomotion between virtual environments, text input, and a design for more embodied desktop-based avatars in virtual immersive environments which have yet to be implemented.

5.2 Final Thoughts

If we shift from thinking about VR using mobile and desktop techniques as our inspiration, to instead thinking about doing everything from a more fundamentally 3D perspective, there's an enormous amount of work to be done. This thesis explores some of the open questions in interaction design for VR, and proposes some possible answers, but the number of things we do with a computer is huge. What we're missing for VR interaction design is a mother of all demos of VR, or a WIMP interface for VR. WIMP proposed a design solution for a problem which humans had never interacted with before– how do we interact with arbitrarily large quantities of information? What we need for VR is a comprehensive solution for a new design space which humans have never interacted with before– arbitrarily large quantities of information displayed in arbitrarily large quantities of space. That solution is not WIMP. Maybe some of it is in this thesis. Maybe it's something we can't possibly imagine yet.

Appendix A

Code Links

1. Bird Experiment 1 and 2 Unity Projects
   https://gitlab.com/AubreySimonson/Bird.git

2. Pockets Unity Project
   https://gitlab.com/AubreySimonson/pockets.git

Bibliography

[1] Virtual real(i)ty, 2020.

[2] Hubs by mozilla, Feb 2021.

[3] Merwan Achibet, Adrien Girard, Anthony Talvas, Maud Marchal, and Anatole Lécuyer. Elastic-arm: Human-scale passive haptic feedback for augmenting interaction and perception in virtual environments. In *2015 IEEE Virtual Reality (VR)*, pages 63–68. IEEE, 2015.

[4] Bilal Alchalabi and Jocelyn Faubert. A comparison between bci simulation and neurofeedback for forward/backward navigation in virtual reality. *Computational intelligence and neuroscience*, 2019, 2019.

[5] Ferran Argelaguet, Ludovic Hoyet, Michaël Trico, and Anatole Lécuyer. The role of interaction in virtual embodiment: Effects of the virtual hand representation. In *2016 IEEE Virtual Reality (VR)*, pages 3–10. IEEE, 2016.

[6] Laura Lynn Arns. A new taxonomy for locomotion in virtual environments. 2002.

[7] MUHAMMAD ASHAR IMTIAZ. The effects of different types of huds on cybersickness-effects of diegetic and non-diegetic displays on cybersickness in virtual reality. 2020.

[8] Takumi Azai, Mai Otsuki, Fumihisa Shibata, and Asako Kimura. Open palm menu: A virtual menu placed in front of the palm. In *Proceedings of the 9th Augmented Human International Conference*, pages 1–5, 2018.

[9] Takumi Azai, Syunsuke Ushiro, Junlin Li, Mai Otsuki, Fumihisa Shibata, and Asako Kimura. Tap-tap menu: body touching for virtual interactive menus. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, pages 1–2, 2018.

[10] Alan Baddeley. Working memory. *Science*, 255(5044):556–559, 1992.

[11] Jeremy Bailenson. *Experience on demand: What virtual reality is, how it works, and what it can do.* WW Norton & Company, 2018.

[12] Marc Baloup, Thomas Pietrzak, and Géry Casiez. Raycursor: A 3d pointing facilitation technique based on raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

[13] Doug A Bowman and Larry F Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. *SI3D*, 97:35–38, 1997.

[14] Doug A Bowman, Donald B Johnson, and Larry F Hodges. Testbed evaluation of virtual environment interaction techniques. *Presence: Teleoperators & Virtual Environments*, 10(1):75–95, 2001.

[15] Doug A Bowman, Christopher J Rhoton, and Marcio S Pinho. Text input techniques for immersive virtual environments: An empirical comparison. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 46, pages 2154–2158. SAGE Publications Sage CA: Los Angeles, CA, 2002.

[16] Doug A Bowman and Chadwick A Wingrave. Design and evaluation of menu systems for immersive virtual environments. In *Proceedings IEEE Virtual Reality 2001*, pages 149–156. IEEE, 2001.

[17] Gerd Bruder, Paul Lubos, and Frank Steinicke. Cognitive resource demands of redirected walking. *IEEE transactions on visualization and computer graphics*, 21(4):539–544, 2015.

[18] Vannevar Bush et al. As we may think. *The atlantic monthly*, 176(1):101–108, 1945.

[19] Jack Callahan, Don Hopkins, Mark Weiser, and Ben Shneiderman. An empirical comparison of pie vs. linear menus. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 95–100, 1988.

[20] Wei Chu. Application of data encryption technology in computer network security. In *Journal of Physics: Conference Series*, volume 1237, page 022049. IOP Publishing, 2019.

[21] Rory MS Clifford, Nikita Mae B Tuanquin, and Robert W Lindeman. Jedi forceextension: Telekinesis as a virtual reality interaction metaphor. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 239–240. IEEE, 2017.

[22] Sebastian Cmentowski, Andrey Krekhov, Ann-Marie Müller, and Jens Krüger. Toward a taxonomy of inventory systems for virtual reality games. In *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, pages 363–370, 2019.

[23] Willams Costa, Luca Ananias, Iago Barbosa, Bruno Barbosa, André De'Carli, Ricardo R Barioni, Lucas Figueiredo, Veronica Teichrieb, and Daniel Filgueira. Songverse: a music-loop authoring tool based on virtual reality. In *2019 21st*

Symposium on Virtual and Augmented Reality (SVR), pages 216–222. IEEE, 2019.

[24] Sasha Costanza-Chock. *Design justice: Community-led practices to build the worlds we need.* The MIT Press, 2020.

[25] Rudolph P Darken. Hands-off interaction with menus in virtual spaces. In *Stereoscopic Displays and Virtual Reality Systems*, volume 2177, pages 365–371. International Society for Optics and Photonics, 1994.

[26] John Dudley, Hrvoje Benko, Daniel Wigdor, and Per Ola Kristensson. Performance envelopes of virtual keyboard text input strategies in virtual reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 289–300. IEEE, 2019.

[27] Jacqui Fashimpaur, Kenrick Kin, and Matt Longest. Pinchtype: Text entry for virtual and augmented reality using comfortable thumb to fingertip pinches. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–7, 2020.

[28] Alex Olwal Steven Feiner. The flexible pointer: An interaction technique for selection in augmented and virtual reality. In *Proc. UIST'03*, pages 81–82, 2003.

[29] John Fillwalk. Chromachord: A virtual musical instrument. In *2015 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 201–202. IEEE, 2015.

[30] Scott W Greenwald, Wiley Corning, and Pattie Maes. Multi-user framework for collaboration and co-creation in virtual reality. 12th International Conference on Computer Supported Collaborative Learning . . . , 2017.

[31] Tovi Grossman and Ravin Balakrishnan. The design and evaluation of selection techniques for 3d volumetric displays. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 3–12. ACM, 2006.

[32] Chris Harrison, Robert Xiao, Julia Schwarz, and Scott E Hudson. Touchtools: leveraging familiarity and skill with physical tools to augment touch interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2913–2916, 2014.

[33] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier, 1988.

[34] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1063–1072. ACM, 2014.

[35] Eric Hodgson and Eric Bachmann. Comparing four approaches to generalized redirected walking: Simulation and live user data. *IEEE transactions on visualization and computer graphics*, 19(4):634–643, 2013.

[36] Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. Digitspace: designing thumb-to-fingers touch interfaces for one-handed and eyes-free interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1526–1537, 2016.

[37] Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Transactions on Graphics (ToG)*, 34(4):1–14, 2015.

[38] Tommi Ilmonen and Markku Reunanen. Virtual pockets in virtual reality. In *IPT/EGVE*, pages 163–170, 2005.

[39] 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. ACM, 1990.

[40] David Johnson and George Tzanetakis. Vrmin: using mixed reality to augment the theremin for musical tutoring. In *NIME*, pages 151–156, 2017.

[41] Robert S Kennedy, Norman E Lane, Kevin S Berbaum, and Michael G Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology*, 3(3):203–220, 1993.

[42] Regis Kopper, Felipe Bacim, and Doug A Bowman. Rapid and accurate 3d selection by progressive refinement. In *2011 IEEE symposium on 3D user interfaces (3DUI)*, pages 67–74. IEEE, 2011.

[43] Nataliya Kosmyna, Franck Tarpin-Bernard, and Bertrand Rivet. Conceptual priming for in-game bci training. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 22(5):1–25, 2015.

[44] Larry Koved and Ben Shneiderman. Embedded menus: Selecting items in context. *Communications of the ACM*, 29(4):312–318, 1986.

[45] Arun Kulshreshth and Joseph J LaViola Jr. Exploring the usefulness of finger-based 3d gesture menu selection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1093–1102, 2014.

[46] Juha Lehikoinen. Virtual pockets. In *Digest of Papers. Fourth International Symposium on Wearable Computers*, pages 165–166. IEEE, 2000.

[47] Arwa Michelle Mboya. The oculus go wasn't designed for black hair, Nov 2020.

[48] Joshua McVeigh-Schultz and Katherine Isbister. The case for "weird social" in vr/xr: A vision of social superpowers beyond meatspace. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–10, 2021.

[49] Mark R Mine. Virtual environment interaction techniques. *UNC Chapel Hill CS Dept*, 1995.

[50] Mark R Mine, Frederick P Brooks Jr, and Carlo H Sequin. Moving objects in space: exploiting proprioception in virtual-environment interaction. In *SIGGRAPH*, volume 97, pages 19–26, 1997.

[51] Pedro Monteiro, Hugo Coelho, Guilherme Gonçalves, Miguel Melo, and Maximino Bessa. Comparison of radial and panel menus in virtual reality. *IEEE Access*, 7:116370–116379, 2019.

[52] Alec G Moore, Michael J Howell, Addison W Stiles, Nicolas S Herrera, and Ryan P McMahan. Wedge: A musical interface for building and playing composition-appropriate immersive environments. In *2015 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 205–206. IEEE, 2015.

[53] Axel GE Mulder. *Design of virtual three-dimensional instruments for sound control*. Simon Fraser University Canada, 1998.

[54] Christian T Neth, Jan L Souman, David Engel, Uwe Kloos, Heinrich H Bulthoff, and Betty J Mohler. Velocity-dependent dynamic curvature gain for redirected walking. *IEEE transactions on visualization and computer graphics*, 18(7):1041–1052, 2012.

[55] Don Norman. *The design of everyday things: Revised and expanded edition*. Basic books, 2013.

[56] Chanho Park, Hyunwoo Cho, Sangheon Park, Young-Suk Yoon, and Sung-Uk Jung. Handposemenu: Hand posture-based virtual menus for changing interaction mode in 3d space. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*, pages 361–366, 2019.

[57] Tekla S Perry and John Voelcker. Of mice and menus: designing the user-friendly interface. *IEEE Spectrum*, 26(9):46–51, 1989.

[58] Markus Petrykowski, Philipp Berger, Patrick Hennig, and Christoph Meinel. Digital collaboration with a whiteboard in virtual reality. In *Proceedings of the Future Technologies Conference*, pages 962–981. Springer, 2018.

[59] Everest Pipkin. This is not my beautiful house: Examining the desktop metaphor, 1980-1995. *continent.*, 8(1):122–129, 2019.

[60] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in vr. In *ACM Symposium on User Interface Software and Technology*, pages 79–80. Citeseer, 1996.

[61] Ivan Poupyrev, Tadao Ichikawa, Suzanne Weghorst, and Mark Billinghurst. Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques. In *Computer graphics forum*, volume 17, pages 41–52. Wiley Online Library, 1998.

[62] Gonzalo Ramos, George Robertson, Mary Czerwinski, Desney Tan, Patrick Baudisch, Ken Hinckley, and Maneesh Agrawala. Tumble! splat! helping users access and manipulate occluded content in 2d drawings. In *Proceedings of the working conference on Advanced visual interfaces*, pages 428–435, 2006.

[63] Oscar Rosello Gil Rosello. *Nevermind: an interface for human memory augmentation*. PhD thesis, Massachusetts Institute of Technology, 2017.

[64] Diego C Ruspini, Krasimir Kolarov, and Oussama Khatib. The haptic display of complex graphical environments. In *Siggraph*, volume 97, pages 345–352, 1997.

[65] Tao Ni Ryan, P McMahan, and Doug A Bowman. Tech-note: rapmenu: remote menu selection using freehand gestural input. In *2008 IEEE Symposium on 3D User Interfaces*, pages 55–58. IEEE, 2008.

[66] Joshua Salyers, Daniel Cliburn, Keely Canniff, and Stephany Barajas. Evaluation of information widgets for a virtual reality serious game. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1627–1632. IEEE, 2019.

[67] Andrés Santos, Telmo Zarraonandia, Paloma Díaz, and Ignacio Aedo. A comparative study of menus in virtual reality environments. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, pages 294–299, 2017.

[68] Ludwig Sidenmark, Christopher Clarke, Xuesong Zhang, Jenny Phu, and Hans Gellersen. Outline pursuits: Gaze-assisted selection of occluded objects in virtual reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.

[69] Mel Slater, Martin Usoh, and Anthony Steed. Depth of presence in virtual environments. *Presence: Teleoperators & Virtual Environments*, 3(2):130–144, 1994.

[70] Marco Speicher, Jan Ehrlich, Vito Gentile, Donald Degraen, Salvatore Sorce, and Antonio Krüger. Pseudo-haptic controls for mid-air finger-based menu interaction. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2019.

[71] Misha Sra. *A framework for enhancing the sense of presence in virtual and mixed reality.* PhD thesis, Massachusetts Institute of Technology, 2018.

[72] Kay M Stanney, Mansooreh Mollaghasemi, Leah Reeves, Robert Breaux, and David A Graeber. Usability engineering of virtual environments (ves): identifying multiple criteria that drive effective ve system design. *International Journal of Human-Computer Studies*, 58(4):447–481, 2003.

[73] Thad Starner, Joshua Weaver, and Alex Pentland. A wearable computer based american sign language recognizer. In *Digest of Papers. First International Symposium on Wearable Computers*, pages 130–137. IEEE, 1997.

[74] Frank Steinicke, Gerd Bruder, Jason Jerald, Harald Frenz, and Markus Lappe. Analyses of human sensitivity to redirected walking. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pages 149–156, 2008.

[75] Frank Steinicke, Gerd Bruder, Jason Jerald, Harald Frenz, and Markus Lappe. Estimation of detection thresholds for redirected walking techniques. *IEEE transactions on visualization and computer graphics*, 16(1):17–27, 2009.

[76] Frank Steinicke, Timo Ropinski, and Klaus Hinrichs. Object selection in virtual environments using an improved virtual pointer metaphor. In *Computer vision and graphics*, pages 320–326. Springer, 2006.

[77] Ivan Sutherland. The ultimate display. 1965.

[78] Dangxiao Wang, Kouhei Ohnishi, and Weiliang Xu. Multimodal haptic display for virtual reality: A survey. *IEEE Transactions on Industrial Electronics*, 2019.

[79] Yanbin Wang, Yizhou Hu, and Yu Chen. An experimental investigation of menu selection for immersive virtual environments: fixed versus handheld menus. *Virtual Reality*, 25(2):409–419, 2021.

[80] Konstantin Wegner, Sven Seele, Helmut Buhler, Sebastian Misztal, Rainer Herpers, and Jonas Schild. Comparison of two inventory design concepts in a collaborative virtual reality serious game. In *Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play*, pages 323–329, 2017.

[81] Bob G Witmer and Michael J Singer. Measuring presence in virtual environments: A presence questionnaire. *Presence*, 7(3):225–240, 1998.

[82] Hans Peter Wyss, Roland Blach, and Matthias Bues. isith-intersection-based spatial interaction for two hands. In *3D User Interfaces (3DUI'06)*, pages 59–61. IEEE, 2006.

[83] Yukang Yan, Chun Yu, Xiaojuan Ma, Xin Yi, Ke Sun, and Yuanchun Shi. Virtualgrasp: Leveraging experience of interacting with physical objects to facilitate digital object retrieval. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2018.

[84] Daniel Zielasko, Marcel Krüger, Benjamin Weyers, and Torsten W Kuhlen. Passive haptic menus for desk-based and hmd-projected virtual reality. In *2019 IEEE 5th Workshop on Everyday Virtual Reality (WEVR)*, pages 1–6. IEEE, 2019.

[85] Daniel Zielasko, Dominik Rausch, Yuen C Law, Thomas C Knott, Sebastian Pick, Sven Porsche, Joachim Herber, Johannes Hummel, and Torsten W Kuhlen. Cirque des bouteilles: The art of blowing on bottles. In *2015 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 209–210. IEEE, 2015.