Title
Simulation and Efficiency: Interaction Design Analysis of Virtual Reality Games

Author
Jing, Ke

Publication Date
2021

UNIVERSITY OF CALIFORNIA,
IRVINE

Simulation and Efficiency: Interaction Design Analysis of Virtual Reality Games

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Informatics

by

Ke Jing

Dissertation Committee:
Associate Professor Theresa Jean Tanenbaum, Co-chair
Assistant Professor Aaron Trammell, Co-chair
Professor Katie Salen Tekinbaş

2021

DEDICATION

I dedicate this work to my wife and my mom for their love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

ACKNOWLEDGEMENTS

Although working on this dissertation is primarily a personal and isolating experience, I could not have produced it without many people's help and support during my graduate study.

First of all, I thank my dissertation committee members: Tess Tanenbaum, Aaron Trammell, and Katie Salen, for pushing and encouraging me to do good work, for teaching me the ways of thinking about things not as problems to be solved, but as phenomena to be understood. I thank Tess for lending me so many insights into digital media, video games, and extended reality, including a framework that I take as a core theoretical framework to start this work. I thank Aaron for introducing me to platform studies and critical game analysis. Thank you for mentoring me with kind words through the dissertation writing process. I thank Katie for giving me a lot of perspectives on design, including the lenses of value tensions in design that I took from your advanced design course applied for this work. I feel grateful for having these three experts on my committee. Their knowledge, wisdom, and professionalism will have a long influence on my future career.

I thank all my professors, mentors, and colleagues: Kurt Squire, Salma Elmalaki, Stacy Branham, Bo Ruberg, Constance Steinkuehler, Melissa Mazmanian, Paul Dourish, Geoffrey Bowker, Neil Young, Mari Kimura, Christian Shelton, AC Mahendran, Matt Dalby, Chris Massie, Tim Kashani, Barry Stout, and more. I thank all my friends, mates, and cohort: Amanda Cullen, Buyun Zhang, Dan Gardner, Je Seok Lee, Lika Liu, Malcolm Liu, Maruf Zaber, Max Collins, Nick Su, Saumya Gupta, Will Dunkel, Xi Chen, Xi Lu, Yao Du, and more. I thank all the organizations, labs, and companies that I worked with: the Department of Informatics in the Donald Bren School of Information and Computer Sciences, the Transformative Play Lab, the CATS lab, LUCI lab, Calit2, Osram Sylvania, and Unity Technologies.

I truly treasure the moments of inspiration from my experiences with each of you, and the wonderful time that we had together shall always remain in my heart. Finally, the special thanks go to my wife, my mom, and family members. Thank you for your love and support! I love you!

VITA

2021	Ph.D. in Informatics, University of California, Irvine
2019	M.S. in Informatics, University of California, Irvine
2016	Advanced Certificate in Engineering, University of California, Riverside
2016	B.E. in Automation, Shenzhen University

FIELD OF STUDY

Game Design, Virtual and Extended Reality, Human-Computer Interaction, Platform Studies.

PUBLICATIONS

Jing, K. (2020). Computing Models of Escape Rooms. *Analog Game Studies*, *VII*(II). https://analoggamestudies.org/2020/12/computing-models-of-escape-rooms/

Jing, K., Nygaard, N., & Tanenbaum, J. (2017). Magia Transformo: Designing for Mixed Reality Transformative Play. *Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play*, 421–429. https://doi.org/10.1145/3130859.3131339

PROFESSIONAL EXPERIENCE

2020	Software Engineer Intern (XR Interaction Toolkit)	Unity Technologies
2019	Software Engineer Intern (XR Interaction Toolkit)	Unity Technologies
2018	Graduate Internship – AR/VR (R&D Projects)	Osram Sylvania

TEACHING ASSISTANT EXPERIENCE

Visual Design Fundamentals, Critical Writing, Advanced Design, Design and Prototype, Game Design, Capstone Game, Human-Computer Interaction, Interactive Technologies

ABSTRACT OF THE DISSERTATION

Simulation and Efficiency: Interaction Design Analysis of Virtual Reality Games
by
Ke Jing
Doctor of Philosophy in Informatics
University of California, Irvine, 2021
Associate Professor Theresa Jean Tanenbaum, Co-chair
Assistant Professor Aaron Trammell, Co-chair

From *Sword Art Online* to *Ready Player One*, people have never stopped envisioning how virtual reality (VR) can transform digital games and potentially everyday life. As VR continually evolves, understanding its interaction design is crucial. In this dissertation, I establish a framework of simulation and efficiency as two fundamental design values for VR game interactions. I theorize the design of VR game interactions as reconciling value tensions between simulation and efficiency through case studies of the state-of-the-art in the first generation commercial VR games. In game studies, simulation often refers to how well game interactions represent their real-world counterparts, and efficiency is often an aspect of usability, playability, and ludic interaction. I contend that simulation and efficiency can be viewed further as duality rather than mapping them to separate continuums and scales as in existing frameworks. I examine the formal elements of the platform, the interaction, and the game context within the interaction design to see how the value tensions between simulation and efficiency can be amplified or reconciled.

My dataset includes hundreds of gameplay screenshots depicting interaction design examples of over 20 semantic actions organized under four categories: 1) Locomotion Interactions, 2) Object Interactions, 3) FPS Combat Interactions, and 4) RPG Combat Interactions. I use the Oculus Rift S headset and motion controllers as the VR platform for my close reading. I analyze several representative examples in each category and map them onto the two-dimensional simulation and efficiency framework for comparison.

The analysis produces new understandings of simulation and efficiency and extends two existing frameworks – 1) interaction fidelity and 2) narrative and embodied interface. I redefine simulation to emphasize the experiential perspective of achieving realism over its predominant formal perspective of being technically realistic. I provide a new perspective for the controversial uncanny valley model in the interaction fidelity framework to reveal the risk of over-simulating the realistic. For efficiency, I propose a hierarchical model to highlight levels of inefficient interactions and the contextual efficiency of achieving the desired game goal. For practical contributions, I analyze the emerging best practices (high simulation, high efficiency) and representative cases of suboptimal design. Finally, I propose several principles and guidelines for future VR game interaction design based on the new simulation and efficiency framework.

1. Background and Introduction

People have never stopped imagining how Virtual Reality (VR) can transform digital games and everyday lives. From the anime *Sword Art Online* to the film *Ready Player One*, we see the future and the potential of VR games. Through full-dive VR interfaces, such as the brain-computer interface helmet and the wearable display and suit (Figure 1), players can feel their presence as virtual characters and experience all kinds of fun in VR.

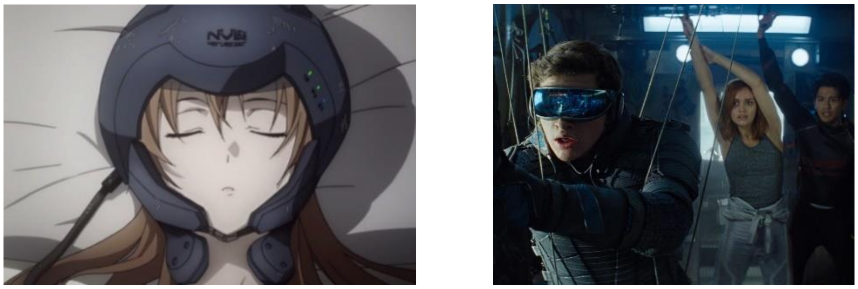


Figure 1. The brain-computer interface in *Sword Art Online* (left) and the wearable display and suit in *Ready Player One* (right).

VR stems from early research in display and tracking systems in the 1960s. Ivan Sutherland, a pioneer of creating VR systems, stated in his lecture "The Ultimate Display":

> "The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal. With appropriate programming such a display could literally be the Wonderland into which Alice walked."
> (Sutherland, 1965)

With the release of first-generation commercial VR devices like the Oculus Rift and the HTC Vive, modern VR games have created many versions of such wonderlands. However, the more we learn about VR, the more challenges and problems we encounter. Some of these problems are purely technical, but many of these problems require new understandings of interaction design.

A VR experience, or "any in which the user is effectively immersed in a responsive virtual world"(Brooks, 1999), involves user dynamic control and interaction. Interestingly, the two images in Figure 1 illustrate two extremes of the experience with VR interactions and interfaces. One controls every virtual action by simply thinking of it through the brain-computer interface (and perhaps this is the most efficient approach). The other is a one-to-one simulation between virtual activities and physical body movements.

As a step towards VR being "the next interaction platform", today's commercial VR devices lie between the two extreme interfaces of simulating physical actions. Figure 2 shows the Oculus Rift S headset and motion controllers. Each motion controller consists of a joystick and five buttons. They simulate a tracked virtual hand, mapping the index finger onto the Trigger Button and the middle finger onto the Grip Button. The controller's spatial tracking mechanism and limited physical inputs support a limited degree of embodiment, with some actions often reductively mapped to simple button presses. Still, current commercial VR platforms are a significant paradigm shift from mainstream desktop, mobile, and console platforms in terms of interaction.

Figure 2. Oculus Rift S headset and motion controllers

For example, in many desktop games, players use the keyboard and the mouse with custom key bindings to move the character and perform various locomotion, combat, and object interactions. The mouse is often used for looking around and point-and-click interactions. Mobile games use the joystick button on the left side to move and buttons on the right side to cast abilities. For gaming consoles, the control layouts and mapping have also been standardized through many iterations since the 1980s. Overviewing these traditional platforms, we can find that the consistent theme for controller inputs is to use the left hand for movement controls and the right hand for looking around and casting abilities.

Figure 3. One of the world's most popular desktop MMORPG, *World of Warcraft* (Blizzard Entertainment, 2004), uses ActionBars to bind abilities to custom keys (left). The Chinese market's most popular mobile battle arena game, *Honor of Kings* (TiMi Studio Group, 2015), uses a left joystick button for movement and right buttons for actions (right).

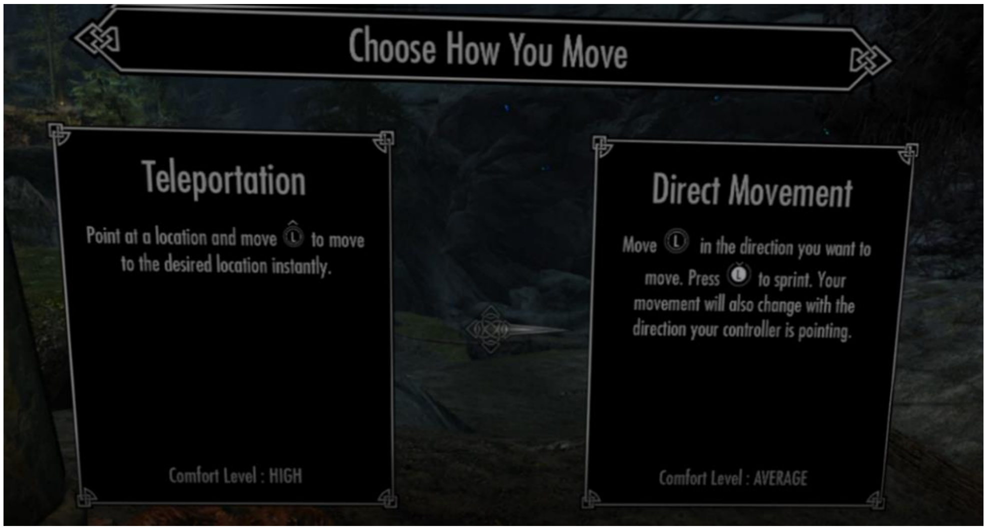
VR interactions often inherit the control literacy from traditional platforms. Still, the new affordances and constraints of VR hardware have opened new design space for player interactions. Before discussing more abstract and theoretical concepts, I will show a few VR game interactions from two of my favorite modern VR games in the following section. Through the examples, I will highlight issues and key ideas in this dissertation, such as simulation, efficiency, game context and the platform.

1.1. Experiencing VR Game Interactions

The best way to introduce VR game interactions is to show them and explain my experience with them. I will go over some basic locomotion and combat interactions in the game *The Elder Scrolls V: Skyrim VR* (Bethesda Game Studios, 2017), or *Skyrim VR* for short, and object interactions in the game *Population: ONE* (BigBox VR, Inc., 2020).

1.1.1. Locomotion and Combat Interactions – the case of *The Elder Scrolls V: Skyrim VR*

Skyrim VR is an open-world role-playing game (RPG) with a medieval setting. As its name suggests, it has the same content as the original game but experienced with VR interactions. The game eases players into VR by setting up its first scene as a tutorial for basic locomotion techniques. For movement options, I could only choose between Teleportation and Direct Movement (Figure 4).



Choose How You Move

Teleportation

Point at a location and move Ⓛ to move to the desired location instantly.

Comfort Level : HIGH

Direct Movement

Move Ⓛ in the direction you want to move. Press Ⓛ to sprint. Your movement will also change with the direction your controller is pointing.

Comfort Level : AVERAGE

Figure 4. Movement options in *Skyrim VR*: Teleportation and Direct Movement

Teleportation uses the "Point and Teleport" technique to reduce the motion sickness[1] that is more often experienced in the direct movement and the walk-in-place approaches (Bozgeyikli et al., 2016). To initiate teleportation in *Skyrim VR*, I pushed the left joystick up to see a projective curve with a cursor on the ground. I could adjust the cursor position by rotating my left wrist up and down, and on releasing the joystick, I teleported to where the curve landed.

Before

After

Figure 5. Teleportation in *Skyrim VR*

---

[1] Motion sickness, or cybersickness, is likely to happen when motion is only sensed by the eyes but not other systems of the body (Reason, 1978, pp. 819–829). It can also happen when signals sent from motion sensing systems are inconsistent with what they normally would be in real-world situations.

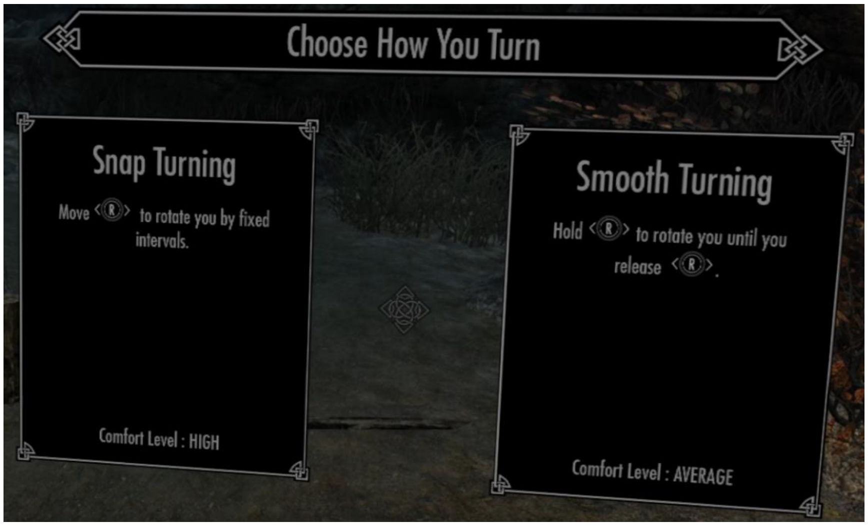
Although Teleportation could be magical and comfortable to perform, I quickly realized that it might not be efficient and ideal in some cases, like melee combats (attacking enemies with hand weapons). Bozgeyikli et al. (2016), who described the Point-and-Teleport technique, acknowledged that its experience might differ in more dynamic, challenging, and high paced scenarios. Teleportation would be a good choice for some tasks like travelling. However, I still prefer the more direct way of moving around, like in conventional games.

With Direct Movement, I find it more intuitive to move around, especially when performing micro-movements in melee combats. I find that I no longer have to shift my attention to the ground before moving, nor do I have to check the surroundings to confirm that I have landed at the correct position. However, sometimes Direct Movement did trigger motion sickness, especially when there were drastic visual changes. However, I gradually got used to it and still favored the naturalness of Direct Movement.

Since players can only choose one of the two movement styles, will there be cases in which one has more advantages than the other? How would the game design of *Skyrim VR* balance the two if it was a more competitive game or like an MMORPG? With the current hardware, many users still may not feel comfortable with Direct Movement. However, it will not be fair to them if their only option - Teleportation leads to suboptimal combat performance. If Teleportation is faster for traveling, or if some fights benefit from blinking (the teleporting effect), players who prefer Direct Movement may feel frustrated for not having a perfect choice.

Then why not have both movement techniques active at the same time? It is technically feasible, but it reveals another design problem of *control schemes*, which I will discuss further in later chapters. If Direct Movement is enabled, the left joystick will be "fully booked". Players cannot have Direct Movement and Teleportation simultaneously because the right joystick and other buttons are also fully assigned to other essential controls. The right-side control mappings follow the convention to cover looking around and performing actions. These controls also face similar trade-offs as those between Teleportation and Direct Movement.

First, pushing the right joystick left/right is mapped to turning left/right. There are also two options for turning – Snap Turning, which turns a fixed angle each time, and Smooth Turning, which is the continuous mode for looking around similar to conventional video games (Figure 6).


Choose How You Turn

Snap Turning

Move ‹R› to rotate you by fixed intervals.

Comfort Level : HIGH

Smooth Turning

Hold ‹R› to rotate you until you release ‹R›.

Comfort Level : AVERAGE

Figure 6. Snap/Smooth Turning in *Skyrim VR*

The Snap/Smooth Turning would presumably share the same trade-offs as Teleportation and Direct Movement. However, I didn't find Snap Turn significantly sub-optimal in its performance of rotating to a specific angle quickly. I could always make micro-adjustments by physically turning my head, thanks to the VR tracking. However, after physical micro-adjustments accumulated for a long time, I would lose the real-world direction, risking hitting my desk or the wall in my play space. This problem will be less frequent if I force my head never to move away from the original safe direction and only use the continuous Smooth Turning to look around in VR. Although Smooth Turning is not the default option, I switched to this interaction technique. I favored how it simulated the realism of continuously turning around and followed the convention in non-VR games. In addition, I increased the Rotation speed so that I could be prepared for what might come in all directions. Surprisingly, the fast-moving animation was not too uncomfortable when most scenes were in large open areas. However, there were few occasions when an enemy attacked from behind, and I had to react with a physical and virtual turning joint.

Turning Mode Smooth
Snap Transition Gradual
Angle Snap Amount
Rotation Speed
Movement Mode Direct Movement
Direct Movement Mode HMD-Relative
Compass Position Low
Main Hand Right

Figure 7. Locomotion setting menu in *Skyrim VR*

It is worth noting that my seated (desktop) VR configuration constrains physical moving (walk-in-place) and turning. Some study suggests that room-scale VR leads to better immersion and performance for some serious tasks (Shewaga et al., 2020). However, the room-scale mode is luxurious and laborsome, and not every player can afford this play style. While some VR simulation games have physical jumping and crunching that benefit from or require the room-scale mode, *Skyrim VR* maps jumping and sneaking to pushing the right joystick up and down, respectively. I would quickly get fatigued and break the immersion if I kept standing for hours to play *Skyrim VR*, let alone physically jump or crunch.

Realistic Swim was an impressive and novel interaction to me. To do that, I need to move my head under the water's surface, hold Left Trigger (LT) and Right Trigger (RT), then move my arms front to back. To stop, I need to move my head above the water. However, after experiencing this realistic simulation a couple of times, I turned off the Realistic option as the "wow, I can do this in VR!" surprise faded. There was no reward for doing such complicated actions for just getting across a river in the gameplay. I stuck to the non-realistic swimming – pushing the joystick to walk in the water.

To put a summary of my experience with the locomotion interactions in *Skyrim VR*, I found their design choices and my preferences were driven by a mixed set of values and factors. While the game by default prioritized comfort (the minimum of usability) for basic moving and turning, I found Teleportation and Snap Turning not realistic and efficient enough in some gameplay contexts. The game does not force players to perform highly simulated interactions for moving, turning, jumping, sneaking, sprinting, and swimming. Still, I enjoyed the more efficient techniques for these actions to deal with fast-paced challenges in my seated VR configuration.

Figure 8. Melee attacking with a two-handed hammer in *Skyrim VR*

As a rare example, the melee attack interaction perfectly marries "simulation" and "efficiency" and suits all gameplay contexts in either seated or room-scale configuration. After pressing the Trigger Button to equip the weapon, I simply swing my hand and arm at the enemy to perform a melee attack. A few extensions to the basic melee attack include holding the Trigger while swinging for a Power Attack, swinging left and right to sweep all enemies in front of me, and Sprint Attack. Again, these actions felt both realistic and easy to perform. The only ironic design was that I did not hold the "two-handed" hammer with both of my hands, but I would rather not.

Figure 9. Archery in *Skyrim VR*

Archery also mimics its real-world action, especially with Realistic Archery. For example, to shoot arrows with a bow, I needed to equip a bow on my left hand and an arrow on my right hand, put the arrow on the bow as it snapped to the bow. Then I could pull the arrow, aim at a target, then release to shoot. With the Realistic mode turned on, I could aim with both hands instead of only with the left hand holding the bow. However, I found it challenging to hit targets without any aiming cursor like those in first-person shooting games. In *Skyrim VR*, since the weapon's positioning becomes flexible in all three dimensions, the aiming mechanic changes to be estimating an invisible ray cast from the bow.

Figure 10. Casting a blizzard spell in *Skyrim VR*

Unlike Archery, two other ranged abilities - magical spells and power shouts are simple one-button-press actions. Casting a blizzard spell, for example, was performed by holding the Trigger for a few seconds. The game animation shows the hand with an increasing magical aura to visualize the casting process. While advanced gestures would be used for casting had the controller support gesture recognition, this hold-to-cast interaction followed the convention of RPG games.

Figure 11. Power Shout in *Skyrim VR*

The immediate casting of Power Shout by pressing the right Grip button was questionable because the Grip is more often used to grab objects in VR games. In this case, the scarcity of available physical inputs forces the design choice of counter-intuitive mapping. I repeatedly mistakenly pressed Grip to cast the shout, only to experience the thrill of a sudden sound and the visual chaos of nearby enemies and allies being staggered.

Figure 12. The Favorites list for quickly switching items in *Skyrim VR*

The limited inputs also lead to *Skyrim VR* not having a shortcut keypress for switching items instantly, like the macros in RPGs. While *Skyrim VR* has a wide variety of weapons and consumables, I must go through the menu UI, which pauses the game, to equip and use them. The Favorites list UI (Figure 12) offers a shortcut menu to streamline what otherwise will require multi-step menu navigation to find and select one item, but it still pauses the game and requires navigating through the list to find the item.

1.1.2. Object Interactions – the case of *Population: ONE*

Besides locomotion and RPG combat interactions, many object interactions that are automatic or "point-and-click" in traditional games become more detailed manual processes in VR. Before diving into the theories and literature of the field, I want to show a couple of object interactions in the VR game *Population: ONE*. In these examples, I highlight how various design aspects, to certain degrees, enable the interactions to achieve both values of simulation and efficiency.

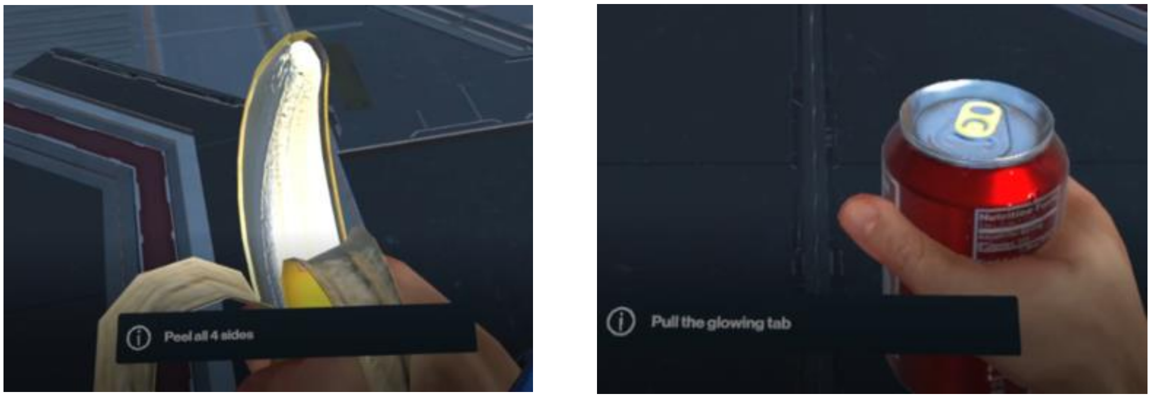
Population: ONE is a VR version of popular first-person shooting (FPS) battle royale games such as Fortnite and PlayerUnknown's Battlegrounds, which involve gun shooting and interacting with objects as their core mechanics.

The first example is the reloading interaction. Reloading usually requires little attention and effort in traditional FPS games. Typically, it is pressing the "R" button on the keyboard and an automatic process whenever the ammo runs out. In VR FPS games, however, the norm for reloading seems to be a laborious process simulating multiple steps: the player has to grab the ammo from the pocket slot, insert the ammo into the gun, then pull the side handle on the weapon to reload. For novice players, finding those slots and precisely performing the actions can become a skill test that can take a considerably long time while in combat.

Figure 13. Reloading in *Population: ONE*

Population: ONE has a couple of designs to facilitate this process. First, as shown in Figure 13, there are highlighted shaders on the handle of the gun and a reloading icon on the aiming cursor as *signifiers* (Norman, 2008; Pearce, 2012). Second, the pulling action does not need to be strictly parallel to the gun. As described in (Krompiec & Park, 2019), the motion controller movement can be projected in the expected reload direction to enhance usability. Although visual highlighting and movement projection are not strictly realistic, they keep the overall sense of simulating the realistic actions and improve the efficiency of performing those actions.

Another example of how VR object interactions reveal the tension between simulation and efficiency also involves precise interactions with small objects. In *Population: ONE*, eating a banana and drinking a soda are ways to recover health. However, as shown in Figure 14, I need to peel all four sides of the banana before moving it close to my mouth to eat it. The problem was that the four sides were so close to each other that I often selected the wrong side. As a result, I either accidentally reversed it and did nothing after a few trials. In contrast, opening the soda was a lot easier because there was only one tab, and it is clearly highlighted.



Peel all 4 sides

Pull the glowing tab

Figure 14. Eating a banana and drinking a soda in *Population: ONE*

It is worth noting in these two cases that my virtual hand *collider* became a double-edged sword. Because the hand collider is probably of a constant size and in the shape of a cube or sphere, it does not have the dedicated two-fingertip controls to precisely open one side among the four at the tip of the banana. However, having only a chunk collider for the hand simplifies pulling the soda tab by making the effective volume bigger than it looks in the game.

From these experiences and thoughts, I gain the motivation to start this dissertation study to systematically examine emerging approaches to design interactions in modern VR games. The following section will situate my work within a scholarly background and formulate orienting concepts and research questions.

1.2. Orienting Concepts and Research Questions

I position this study as an inquiry into the two fundamental *values* that can be pursued when designing interactions in VR games: simulation and efficiency. To approach this, I consider two things. First, I review existing theories of simulation and efficiency in designing game interfaces and interactions. Second, I study emerging practices of VR game interactions, and then I use the theories and practices to explain and understand each other. Thus, this dissertation aims to extend both the theories and design knowledge of VR game interactions.

The theoretical frameworks that I use to build my work and concepts of simulation and efficiency are primarily from the two frameworks: 1) Narrative and Embodied

Interface and 2) Interaction Fidelity. I briefly introduce them here and will discuss them in more detail in Chapter Two of this dissertation.

Narrative and Embodied Interface is a framework to understand and resolve the potential disconnection between the experience of narrative and game interfaces (Tanenbaum & Bizzocchi, 2009; Bizzocchi et al., 2011).

Tanenbaum and Bizzocchi (2009) provide two scales that can be mapped onto the concepts of simulation and efficiency. One scale is the *functional narrativization* of the controller, ranging from Literal, Metaphorical to Abstract. This scale reflects simulation in terms of the degree to which the game controller reproduces the "movement, gesture and functionality of its real world referent" (2009). Another scale is the *Ludic Efficiency*, which they defined as "the extent to which an interface device eases or hinders the player's attempt to perform any given operation within the game" (2009). This definition almost perfectly works for my discussions on efficiency for VR interactions, except that the subject should be an interaction instead of an interface device or, more specifically, the design of an action to be performed in VR using the motion controllers.

Bizzocchi et al. (2011) further identified a pair of design strategies, behavioral mimicking and behavioral metaphor, that reinforce the concept of simulation and narrativization. In these two strategies, the player's physical behavior of operating the interface to perform gameplay interactions can either completely mimic real-world behavior (Behavioral mimicking) or metaphorically suggest a connection to it (Behavioral metaphors). The key difference is "the degree of *fidelity* between the functioning of the interface and the activity which it represents" (Bizzocchi et al., 2011).

McMahan uses the term interaction fidelity (IF) to describe “the objective degree of exactness with which real-world interactions can be reproduced” (McMahan, 2011). The Framework for Interaction Fidelity Analysis (FIFA) offers more systematic measures of IF, including body segments, motions, forces, and the transfer function of control (McMahan, 2011; McMahan et al., 2016). Although a few studies on VR interactions have explored IF, a dominant conclusion is that both high and low IF lead to better performance and player experience, while the moderate IF often gives the worst results but in some cases can suffice (McMahan et al., 2012, 2016; Nabiyouni et al., 2015; Bhargava et al., 2018; Rogers et al., 2019). The high IF is believed to render better immersion and presence because the interaction is more natural and realistic; the low IF is better for usability concerns such as efficiency and familiarity with traditional platforms. Therefore, interaction fidelity is modelled as an "uncanny valley" of VR interactions (McMahan et al., 2016).

When I try to apply these theories in the canon to a number of examples of interactions that I experience in modern VR games, I find new gaps and even contradictory outcomes. As a starting point, I articulate a set of orienting concepts, for which I will provide more evidence in Chapter Two and later analysis chapters of this dissertation.

1. Extending simulation and efficiency as a duality: Existing theories on game interface and interaction design view simulation and efficiency as two separate continuums and overemphasize the value of simulation and its fidelity. However, examining interactions in VR games can reveal the duality of the two values and cases of their potential *trade-offs* or *tensions* in an extended two-dimensional frame, which I will illustrate in section 2.1.3 and 2.3.4.

2. Rethinking Interaction Fidelity (IF) in modern VR games: The IF "uncanny valley" is a reductive model based on evaluations of VR prototypes in a few lab studies. Current research into IF often misses framing the duality of simulation, as reflected in high IF and efficiency as reflected in low IF. The moderate IF is overwhelmingly devalued. The model does not fully explain the complex design space of modern VR games, where most interactions span the medium level of IF.

3. The game context and technical aspects of the platform influence the design of VR interactions. While many studies focused on single variables of input techniques, particularly locomotion techniques, few have systematically examined other VR interactions and related elements, such as visual, audio, haptic feedback, signifiers, controller scheme, game challenges and narrative.

4. Dataset for VR interactions: While large, public, and organized dataset have boosted many fields of computing, such as machine learning, few such dataset of VR interactions exist. Current research into VR interactions often uses prototypes or analyses a small number of examples. As a result, their results have limitations such as generalizability, external validity, and replicability.

For this dissertation and future research on VR interactions, I collected a dataset consisting of short video clips, images, and text descriptions of commonly performed actions in modern VR games. Based on this dataset and the orienting concepts, I explore the following research questions (RQs).

The first question comes from applying examples of VR game interactions to explain and understand existing theoretical frameworks on the concepts of simulation and efficiency. It seeks to extend the frameworks in Narrative and Embodied Interface and Interaction Fidelity by considering the duality of the two design values:

RQ1: How do VR game interactions reveal value tensions between simulation and efficiency?

The second question comes from applying the new framework of simulation and efficiency as lenses to explain and understand the design space of VR game interactions:

RQ2: What design strategies for VR game interactions can we learn with the new understanding of simulation and efficiency? Specifically, in what ways and to what extent do VR games reconcile or amplify value tensions between simulation and efficiency?

1.3. Dissertation Outline

I provide this guide here to help readers navigate the work.

Chapter 1: Introduction

The first chapter that you are currently reading introduces VR game interactions briefly and the core ideas as the research motivation.

Chapter 2: Research Foundations and Literature Review

The second chapter is an extended look at the foundations and related literature of this research. It starts with design values and value tensions as lenses for game interaction design. The rest of the literature review is broken into three subsections:

2.1. Simulation and Efficiency: This section explores the canonical work and theoretical frameworks around simulation and efficiency in game studies. It details the narrativized interface framework and the interaction fidelity framework that this work aims to extend in theory.

2.2. VR Interaction Techniques and Related Design Aspects: This section explains additional design aspects and lenses that I bring to describe VR game interactions and their formal elements.

2.3. Platform Studies and My Previous Work on Extended Reality: This section gives examples of my related work on platform studies for extended reality. These examples help clarify my interest in examining how the VR platform affords or constrains interaction design.

Chapter 3: Data and Methods

This chapter describes the VR Game Interaction dataset, the data collection of interaction design examples from popular commercial VR RPG and FPS games, and the close reading method and the analytic lenses that I will use to study the dataset.

Chapter 4-7: Case Studies of VR Game Interactions

Each of these chapters analyses one category of the VR game interaction design examples in the dataset: Chapter 4 - Locomotion Interactions, Chapter 5 - Object Interactions, Chapter 6 - FPS Combat Interactions, and Chapter 7 - RPG Combat Interactions, drawing on simulation and efficiency as primary lenses and other lenses described previously.

Chapter 8: Discussion and Conclusions

This chapter returns to the analytical lenses and research questions to discuss theoretical contributions and practical contributions. The theoretical contributions involve new understandings of simulation and efficiency as design values from examining the VR game interaction design cases and how these new understandings extend current frameworks in interaction fidelity and narrative and embodied interface. The practical contributions include emerging best practices, representative examples, and design principles and guidelines based on the simulation and efficiency framework. The chapter concludes with a discussion on the platform aspects of VR and the future directions for this research.

2. Research Foundations and Literature

There are many areas, fields, and perspectives related to game studies and interaction design. In this dissertation, I take the following specific positions to study VR game interaction. Firstly, I believe that game design is worth studying for its own sake, and by analyzing the start-of-the-art of design practices, we can gain knowledge of game design and development.

Second, interaction design is a crucial aspect of game design. Interactions can connect Mechanics (rules), Dynamics (what players do), and Aesthetics (how players feel), which are the three components in the MDA framework, a formal framework for game design (Hunicke et al., 2004). Interaction embodies the rules of performing gameplay actions, associates with the inputs of the platform, and involves feedback from the game system. All of these elements can influence the player experience.

Third, game designers can approach interaction design through Aesthetics by translating specific *values*. Using values as a lens for design is a common method in design-related fields, such as the *Value Sensitive Design* (VSD) approach (Friedman, 1996; Friedman et al., 2008) in the field of Human-Computer Interaction (HCI), and the broad conception of *design values* in architecture and industrial design (Holm, 2006). According to Friedman et al. (2008), one way to investigate VSD is to focus on how the technology or mechanism, in this case, VR game interactions, supports or hinders specific values. Aesthetic design values are often reflected in the form of the artefact and the experience with it (Holm, 2006). Furthermore, values often provoke controversy, and a common question of VSD is: How to prioritize competing values in design trade-offs, such as between individual values and usability concerns? In this dissertation, I use the term *value tensions* to frame the trade-offs among competing and diverse values (Miller et al., 2007).

Game scholars have long been interested in and applied design values to study video game design. Zimmerman uses *play values* - "the abstract principles that the game design would embody" to brainstorm ideas and guide the iterative design process (Zimmerman, 2003). He considers game design as a "second-order" design problem, in which designers only create the rules and systems to structure the play experience but not *directly* design it. Until players participate and engage in the game, there is no way to know their experience. Values can be a universal language and bridging point for designers to communicate with players. Designers *embed* values into the rules and systems of the game, hoping that players can experience those values. However, Flanagan et al. suggest that in some cases, such as design choices on the functional components of the game, values introduced by designers can conflict with user values and even broader goals of the project (Flanagan et al., 2005). Apart from delivering a specific experience, many games are intentionally created to reflect broader social, cultural, political, ethical, gender and race values, and game scholars have critically studied these topics, to name a few (Flanagan, 2009; Sicart, 2011; Flanagan & Nissenbaum, 2014; Macklin & Sharp, 2016; de Smale et al., 2019; Ruberg, 2019; Trammell, 2020).

Notably, Flanagan and Nissenbaum's book "Values at Play in Digital Games" (2014) formulates a heuristic framework for designers to approach values in game design. Their heuristics include three components.[2]

> *Discovery*: "Discovery involves (1) locating the values that are relevant to a given project and (2) defining those values within the context of the game."
>
> *Implementation*: Implementation includes translating values into game elements—including specifications, graphics, and lines of code. The heart of design, it is the process of realizing values in terms of the basic practical elements of a game."
>
> *Verification*: "Verification requires establishing the validity of the designer's efforts to discover and implement values. Verification is a form of quality control." (p. 75)

Flanagan and Nissenbaum further suggest four sources to locate values in the Discovery process.

---

[2] Flanagan and Nissenbaum call them “components” instead of “steps” to emphasize the iterative design process.

> 1. *Key actors:* "The people involved in creating the game."
> 2. *Functional description:* "The explicit statement describing the game."
> 3. *Societal input:* "Cultural contexts, standards, and other external factors bearing upon the game."
> 4. *Technical constraints:* "The software, hardware, and other game elements that together constitute the game." (p. 80)

Besides discovering the values, these sources are helpful to evaluate how values are achieved through case studies. In my study, I will focus on the *Technical constraints* among these four to examine how they reveal, amplify, or reconcile the value tensions of simulation and efficiency in VR game interactions.

The technical constraints closely connect to *Implementation*, which is essential because it *translates* values *from the Aesthetics level down to the game elements at the Dynamics or the Mechanics level, as noted at the beginning of my discussions on design values. On Implementation,* Flanagan and Nissenbaum provide two supplemental heuristics:

> 1. *Pay systematic attention to a game's elements.* "In this process, designers consider the full range of a game's elements, such as narrative, character representation, game actions, and even the substrate of game engines and hardware."
> 2. *Consider what you are trying to achieve and how your game conveys values to players (and potentially others).* (p. 101)

The wide range of game elements shows the complex nature of game design, where almost all game elements can be freighted with values. Macklin and Sharp also outline several game elements to establish design values: experience, theme, point of view, challenge, decision-making, skill, strategy, chance, uncertainty, context, and emotions (Macklin & Sharp, 2016).

While most of these works have broadly and critically looked at sociocultural values by analyzing the overall game design in various cases, this dissertation scopes two fundamental design values and VR game interactions. Why do I focus on VR game interactions? First, media scholar McLuhan's famous quote, "The medium is the message" (McLuhan, 1964), implies that there are unique values that may become salient in VR as a new interactive medium. Secondly, game interactions connect various abovementioned game elements, such as inputs, game actions, feedback, and gameplay contexts, where we can systematically examine design values and their trade-offs. The point of applying value-based analysis is less about revealing the values themselves and arousing public attention to them than understanding the design space and gaining design knowledge for VR games.

The rest of this chapter is divided into three categories. The first category extends my brief introduction to the theoretical frameworks that conceptualize the two design values – simulation and efficiency – within the design of game interfaces and interactions. The second deals with related work on VR interactions that use or reflect those theoretical perspectives and relevant aspects of game design. In the third category, I reflect on my previous projects on *Extended Reality (XR)* from the Platform Studies perspective, discussing how the technical constraints of spatial computing platforms can affect the design values. I will also introduce the Oculus Rift S platform that I will use in this study. Finally, I close this chapter by discussing how I will use these areas as *analytical lenses* to investigate VR game interactions.

2.1. Simulation and Efficiency

Simulation is like a grand narrative of VR and video games in general. The name "virtual reality" suggests that it is a simulation of the real world, and so would people think its interactions to be simulations of physical activities. The term "simulation" is broadly used within game studies to describe video games. "All computer games are simulations" (Parker & Becker, 2013). When comparing games and simulations, Crawford (1984) wrote:

> "A simulation is a serious attempt to accurately represent a real phenomenon in another, more malleable form. A game is an artistically simplified representation of a phenomenon. The simulations designer simplifies reluctantly and only as a concession to material and intellectual limitations. The game designer simplifies deliberately in order to focus the player's attention on those factors the designer judges to be important. The fundamental difference between the two lies in their purposes. A simulation is created for computational or evaluative purposes; a game is created for educational or entertainment purposes. Accuracy is the sine qua non (indispensable factor) of simulations; clarity the sine qua non of games. A simulation bears the same relationship to a game that a

> technical drawing bears to a painting. A game is not merely a small simulation lacking the degree of detail that a simulation possesses; a game deliberately suppresses detail to accentuate the broader message that the designer wishes to present. Where a simulation is detailed a game is stylized." (Crawford, 1984, p. 8)

Juul (2005) pointed out that

> "Games are often stylized simulations; developed not just for fidelity to their source domain, but for aesthetic purposes. These are adaptations of elements of the real world. The simulation is oriented toward the perceived interesting aspects of soccer, tennis or being a criminal in a contemporary city." (p. 172)

Bogost (2006) stated that "A simulation is a representation of a source system via a less complex system that informs the user's understanding of the source system" (p. 98). Furthermore, "No simulation can escape some ideological context" (p. 99). To Bogost, there is always a difference between the simulation and its source system, reflecting the creator's subjective design choices.

However, the measurement for simulation or its difference with the simulated counterpart seems still vague from these discussions. Is "accuracy", "realism", or "complexity" the right way to describe the level of simulation? Those words can still be subjective and sometimes difficult to describe. What frameworks can designers apply to understand simulation in the design of game interfaces and interactions?

Breaking away from realism, Dormans uses "*Indexical*" and "*Symbolic*" to describe different levels of simulation beyond Iconic (realistic) simulation in games (Dormans, 2011). His point of simulation is that games' strength is not about accurately modelling fantasy worlds; it is about using *simple* rules to capture complex systems and their overall dynamic behavior. He uses the game *Diablo* (Blizzard Entertainment, 1998)'s inventory interface as an example of *Indexical Simulation*. As shown in Figure 15, the interface uses simple sizes and slots that objects can take to simulate their actual dimensions and weight and the behaviors of managing the bag. Rolling the dice to simulate the multitude of fights and battles in many board games is an example of *Symbolic Simulation*, which Dorman argues is a further step away from realistic modelling. This simulation retains the randomness of fights and the strategies player will take but reduces the detailed actions, which are often not the determining factor. Although Dorman sets three different levels of simulation, his criteria are still based on realism and are discrete labels, as he acknowledges that some examples fall in between the categories. They also loosely connect to 3D interactions.

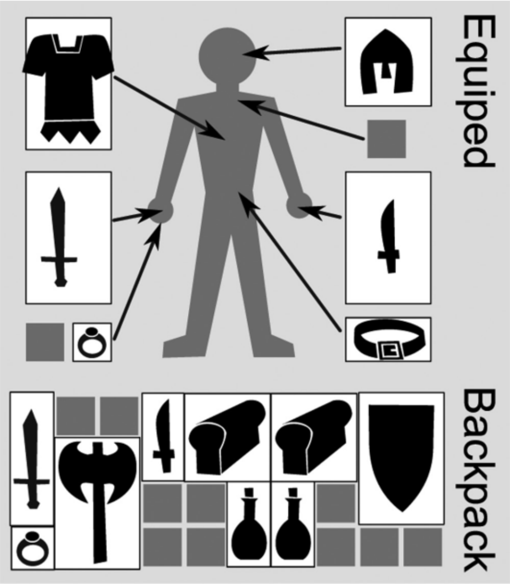


Figure 15. A schematic sketch of *Diablo*'s inventory interface, from (Dormans, 2011)

It is worth noting that Dorman's emphasis on using simple rules to capture complex dynamics also reflects the value of efficiency. In those examples of indexical and symbolic stimulations, the game design intentionally trades simulation details for convenience. Game actions usually take much less time to perform than their simulated counterparts. Achieving the right level of efficiency is a reason why games are fun. According to the flow theory (Csikszentmihalyi, 1990), if an interaction is too challenging, it can make players feel frustrated; if it is too simple, it can lead to boredom. There is an optimal channel between frustration and boredom where game interactions are either too challenging or too simple to keep players in the "flow" state.

Efficiency has a standard definition as an aspect of usability and is indicated by task completion time and learning time[3]. A review of early experimental studies on information-related tasks suggests that efficiency, in general, has weak correlations with other usability aspects such as effectiveness and satisfaction, and they each should be considered independently (Frøkjær et al., 2000). Also, I choose to focus on efficiency instead of including other usability aspects because efficiency is more straightforward to measure and more relevant to game interactions. Game study scholars have applied the well-known concepts of flow and usability to the specific domain of video game design, and efficiency remains an important aspect. For example, Järvinen et al. (2002) establish the model of *playability* and relate its *functional component* to the efficiency of how well the input and output mechanism can achieve gameplay requirements. They argue that functional playability is a precondition of the flow experience. Rouse (2001) suggests that the input and output systems determine the game's learning curve, which relates to efficiency. The Gameflow model maps elements of flow onto specific game literature and provides detailed evaluation criteria for each (Sweetser & Wyeth, 2005). The following are some measures that reflect the concept of efficiency in their descriptions (the original model has more criteria under each element):

Concentration

- games should quickly grab the players' attention and maintain their focus throughout the game
- players shouldn't be burdened with tasks that don't feel important
- games should have a high workload, while still being appropriate for the players' perceptual, cognitive, and memory limits
- players should not be distracted from tasks that they want or need to concentrate on

Player Skills

- game interfaces and mechanics should be easy to learn and use

---

[3] The International Organization for Standardization (ISO) defines efficiency as an aspect of usability. Efficiency is "the relation between (1) the accuracy and completeness with which users achieve certain goals and (2) the resources expended in achieving them. Indicators of efficiency include task completion time and learning time" (International Organization for Standardization, 1998).

Control

- players should not be able to make errors that are detrimental to the game

Clear Goals - Games should provide the player with clear goals at appropriate times

Feedback - Players must receive appropriate feedback at appropriate times

In a nutshell, simulation and efficiency have long been considered in HCI and game studies, and *complexity* seems to be their conceptual meeting point. On the one hand, for simulation, complexity involves accuracy, realism, style, fidelity, and the creator's ideology (Crawford, 1984; Juul, 2005; Bogost, 2006; Dormans, 2011). On the other hand, for efficiency, complexity translates to time and easiness. However, few studies on designing game interfaces and interactions have provided measures for complexity.

Furthermore, while many scholars have discussed simulation and efficiency separately, few critically look at them as *duality*. Indeed, each is important to the design of game interfaces and interactions in general, but when it comes to VR, there is much to understand in how they may become competing values and what the strategies are to deal with them.

The following two sections review two groups of theoretical frameworks that are particularly relevant to this study. They are the core theories that my work is built upon and seeks to extend.

2.1.1. Narrative and Embodied Interface

In the early days of game studies, many video games were simple text-based simulations. Game scholars debated whether the nature of video games was just a form of narrative and whether game studies would be a new branch of narratology. Similarly, the concepts of simulation and narrative share a lot in common. In terms of their definitions, game scholars often describe simulation as a stylized representation of a source system or a phenomenon (Crawford, 1984; Juul, 2005; Bogost, 2006); Narratives are also stylized representations of actual or imaginary events. In addition, both video game simulations and digital narratives are computer interactions that include interactors and actions. They both require inputs, processing, and feedback. Also, simulations in games are like narratives in terms of the experience they can create. They both can be persuasive and immersive (Bogost, 2007; Mäyrä & Ermi, 2011). This section will not continue to discuss "ludology vs narratology" or "simulation vs narrative". By pointing out these conceptual similarities, I intend to provide some background for and support the following theoretical frameworks using *narrative* as a lens to understand the design of game interfaces and the value of simulation.

The narrative and embodied interfaces frameworks were established in Tanenbaum and Bizzocchi's study of the music simulation game Rock Band (Tanenbaum & Bizzocchi, 2009). Rock Band's controllers – a microphone, a drum set, and a guitar – are examples of novel embodied and gestural game interfaces surging at that time before the arrival of commercial VR gaming devices. In HCI, the term "Embodied" is widely used to emphasize being aware of bodily actions and the context. Tanenbaum and Bizzocchi discussed three perspectives of interactions to understand the role of these interfaces in Rock Band's design and experience: the *ludic*, the *kinesthetic*, and the *narrative*.

2.1.1.1. Ludic Efficiency

Firstly, their ludic perspective derives from Csikszentmihalyi's notion of challenge and flow (Csikszentmihalyi, 1990) to define *ludic efficiency* as *the extent to which an interface device eases or hinders the player's attempt to perform any given operation within the game* (Tanenbaum & Bizzocchi, 2009)*.*

With minor adaption, this definition can be perfectly applied to describe the design value of efficiency for VR game interactions. Because in an immersive VR game, players are less aware of the "interface device" - the motion controllers - than their virtual representations in the game. It is what the motion controllers *simulate* in the game context that players directly see and interact with. In many cases, the motion controllers may initially become empty virtual hands. However, what matters more in VR games is what the

Oculus VR cheers players at the end of their setup tutorial - "Now let's see what your virtual hands can do!" Indeed, from grabbing an item, climbing a wall, to equipping a sword and a shield, casting a spell, and firing a shotgun, it is how we perform these actions that require the value of efficiency. Therefore, my adaption of Tanenbaum and Bizzocchi's definition of ludic efficiency to define the design value of efficiency in VR games would be –

Efficiency is the extent to which the interaction design eases or hinders the player's attempt to perform the game action.

Note that the "interaction design" includes the game action but refers to the designed course of performing the action, which involves the controller inputs, bodily movements, UI interfaces to facilitate the action, and feedback from the game system. Therefore, to measure or evaluate the degree of efficiency, I will also consider the time to perform and complete the action, the success/error rate, the design of in-game interfaces and feedback, etc.

Besides the definition, Tanenbaum and Bizzocchi (2009) also demonstrate the scale from *simplistic* to *problematic* where they mapped the three controller interfaces in Rock Band. For this discussion, I add VR interfaces onto their ludic efficiency axis (see Figure 16).

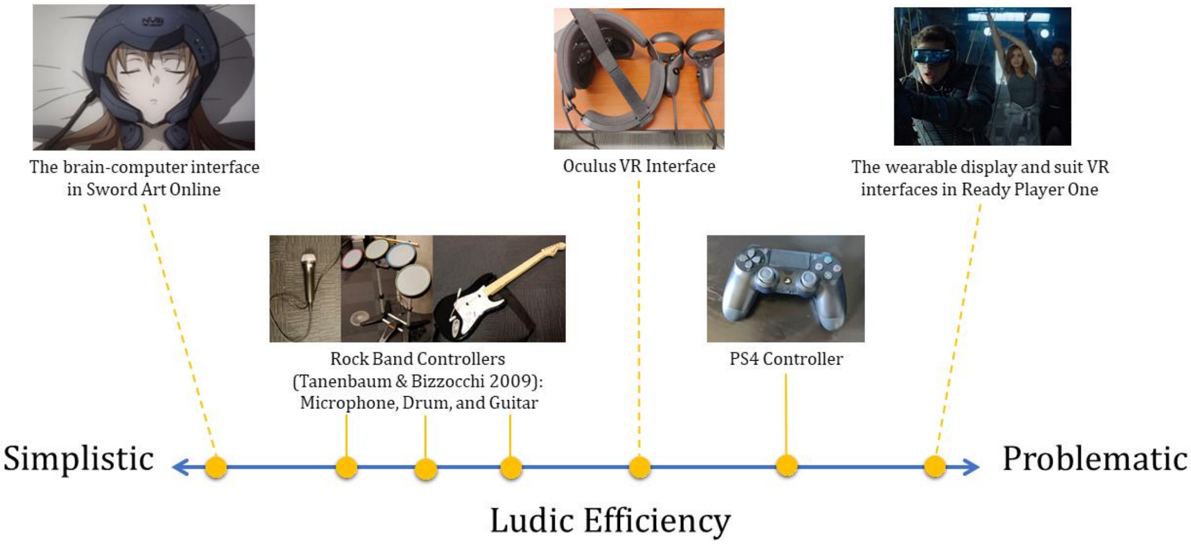


Figure 16. The Ludic Efficiency Scale from Simplistic to Problematic, adapted from (Tanenbaum & Bizzocchi, 2009) to add examples of VR interfaces[4]

The original version only maps the Rock Band controllers – microphone, drum, and guitar and a conventional PlayStation controller onto this scale, the two examples shown at the lower level of the layout. Based on the definition of ludic efficiency, the microphone controller is the most simplistic. It is a *direct simulation* of the physical counterpart - it does not require any learning if one has ever used a microphone. Also, there is no additional mediation between the player making a voice and the interface receiving the audio signal. As a comparison, the drum and guitar controllers are more problematic and inefficient than the microphone. They contain *additional mediation,* such as the color-coded buttons and their positions. Thus, they force players to learn *specific skills* to master their uses. However, they are still less problematic than the conventional game console controller, whose uses and control mappings often require a specialized tutorial at the start of every game.

Based on this reasoning, it is interesting to explore how VR interfaces fit on this scale. However, I find it unconvincing to rank these interfaces without knowing what *specific interactions* and the *contexts* they are to be compared. Unlike the highly specialized Rock Band controllers, the VR interfaces are designed to be *general* game controllers like the Playstation controller. As such, they should be on the problematic side. However, based on the practice of operating them, the envisioned BCI controller is the most simplistic and efficient because it automatically turns thoughts into actions. In contrast, the fully-geared VR suit seems the most problematic because it requires executing everything physically with a complicated setup.

Regarding the commercial (Oculus) VR controller, which is the focused platform of this study, I place it between the PlayStation and the Rock Band controllers because it combines the features of both. As a general controller, it can also simulate many physical instruments through kinesthetic interactions, which we will discuss later. Again, I argue that the evaluation of interactions should depend on specific interactions, like what Rock

---

[4] The mappings of the envisioned and the VR interfaces on this scale are subjective to the context of their interacitons. Unlike the Rock Band controllers whose contexts are known, the three added examples are based on the imagined contexts of performing the interaction with full-body physical behavior (the Ready Player One interface), with only the head and hands (the Oculus VR interface), and with only the brain control (the Sword Art Online interface).

Band is for playing a simulated music concert. The assessment should not be on the interface alone but on how efficiently the *interaction* is designed to be performed *with* the specific interface. For movement interaction, for example, the joystick design on the Oculus motion controller is quite efficient. It is like muscle memory for skilled players. It is probably more efficient than the physical walk-in-space approach. Even the BCI may still require advanced signal processing and suffer from errors to translate movement controls. Similarly, I would also argue that the microphone interface can be inefficient because it is hard to master the voice input to find the right pitch without tangible visualized discrete states such as hitting color-coded buttons and moving the joystick.

The next notion in ludic interaction – *granularity* addresses the interface's technical specifications that can influence ludic interactions. Granularity translates to temporal resolution, or the sample rate of the interface state in *Rock Band* (Tanenbaum & Bizzocchi, 2009). This ludic dimension can significantly constrain the player's action and performance.

In the case of VR games, granularity can influence efficiency as well. However, I think it is more useful to translate granularity to the trigger condition of the game action rather than hardware specifications. Commercial VR devices have reached sample rates high enough beyond what the authors mean by "perceptual threshold" (2009). They are responsive and can accurately represent their position, orientation, and inputs, and players will not perceive a lag between their inputs and the display in most cases. Still, the trigger condition of game action can be strict *by design*, resulting in low efficiency. For example, trigger conditions in VR are often based on collision. The collision can be too delicate to trigger the intended action in some cases. Also, when multiple small colliders are close to each other, the chances are high that the interactor may collide with the wrong target, as I discussed in the introductory cases of consuming the banana.

2.1.1.2. Kinesthetic Interaction, Affordance, and Embodied Context

Their second perspective, kinesthetic, concerns performing bodily actions afforded by the interfaces (Tanenbaum & Bizzocchi, 2009). This perspective less directly relates to simulation or efficiency, but they further discussed several design perspectives and aspects that can reflect the two values.

One design perspective related to kinesis and embodiment is Norman's conceptual models of *affordance*, *constraints*, and *mappings* (Norman, 1988). Affordance and constraints refer to the perceived material properties that afford or constrain the designed operations of the interface.

Take the door and door handle as the classic examples of affordance and constraints. The size of the door affords the action of passing human bodies through it and constrains bodies and objects that are too tall or too wide. A poorly designed door handle that affords both pulling and pushing may confuse first-time users whether the direction of opening the door is to pull or push it. In contrast, a well-designed door handle can constrain one of the actions to prevent the wrong action.

Mapping refers to how the design suggests possible operations and can be revealed through many properties of the interface, such as position and direction. For example, the joystick on the VR motion controller shows a mapping to the character's movement direction. Likewise, the movement of the motion controllers maps directly to the movement of virtual hands in the game. Although affordance, constraints, and mapping are initially derived from the design of physical interfaces, they can be directly applied to the virtual domain to evaluate simulation and efficiency.

Besides considering interfaces as tools, our evaluation can look at the *context* of the game interaction. Each time players perform the same game interaction, the context may be different. Take reloading a gun as an example. Whether players are under time urgency will engender considerable variance in the value of efficiency. Therefore, it is essential to consider the game context when judging the design values of any game interaction.

Using Dourish's notion of context as *socially constructed* (Dourish, 2004), Tanenbaum and Bizzocchi (2009) described the *Rock Band*'s social experience as a simulated live music concert, where players help each other complete the show and engage with non-player spectators. This notion of embodied context can be applied to analyze VR interactions. The context is the meaningful gameplay scenarios where players engage with. For example, they may interact with other players in a multiplayer game, or with non-player characters (NPCs), or with meaningful objects. The physical play area setting –

room-scale or seated – is also a contextual factor, despite not being part of the virtual environment.

While Tanenbaum and Bizzocchi (2009) also borrow the *Viewpoints* framework (Bogart & Landau, 2005) from dance and theater to describe the kinesthetic perspective, I treat these as additional places to locate the design values but will not discuss them in detail. The viewpoints include Tempo, Duration, Kinesthetic Response, and Repetition, as in the Time category, and Shape, Gesture, Architecture, Spatial Relationship, and Topography, as in the Space category.

In summary, the kinesthetic perspective connects to *affordance*, *constraints*, *mappings*, *contexts* of gameplay scenarios, and physical settings. Moreover, they provide useful design aspects to locate the values and conceptual models to evaluate them.

2.1.1.3. Narrativized Interface, Behavioral Mimicking and Behavioral Metaphor

The narrative perspective directly maps to simulation and highlights a measurement scale based on how metaphorically the interface functions as the physical counterpart.

In Bizzocchi's early work, he identified two ways of enhancing narrative experience in designing a game interface – modifying the interface's *look* and *functionality* (Bizzocchi, 2003). Lin further proposed six design strategies for narrativized interface design (Lin, 2007), which then further refined by Bizzocchi, Lin, and Tanenbaum into the following four: 1) The 'look and feel' of the interface, 2) narrativized perspective, 3) Behavioral mimicking and Behavioral metaphor, and 4) 'bridging' and mixed-reality interfaces (Bizzocchi et al., 2011). This section reviews these strategies in detail and discusses how I will adapt them to evaluate the value of simulation in VR game interactions.

The 'look and feel' is an obvious strategy that achieves a sensory matching between the game interface and its real-world referent. The notion of *iconic simulation* (Dormans, 2011) echoes this strategy. The specialized controllers in Rock Band all iconically match the 'look and feel' of their real-world counterparts. Besides the visual and audio aesthetics, this strategy also deals with how *feedback* reinforces the simulated narrative experience. For example, in their analysis of a driving wheel controller, Bizzocchi et al. wrote:

> "The feedback includes the simulation of the effects of momentum, gravity and friction on the car's steering wheel and tires. For example, when driving on dirt, the unevenness of the surface causes the steering wheel to shake and rumble… This also reinforces narrative immersion into the storyworld of driving…" (Bizzocchi et al., 2011)

However, richer feedback does not equal a higher value of simulation. It can be the opposite, especially when simulation is evaluated based on "*realistic"*. For example, the aiming cursor icon and the game metrics[5] of health and ammo are poor simulations of real-world shooting, in which you do not see those visuals in front of your eyes directly. On the other hand, however, they may add to efficiency and the ludic experience. Still, graphical and other sensory expressions are rich channels to reflect simulation, though many VR game interactions may already take it for granted.

From an interaction design point of view, there is more to be learned in how VR interactions simulate functionality than how they simulate sensory expressions. For example, how do VR game interactions achieve the functions of a simulated physical interface while reducing the *fidelity* of operating it? This question leads to reviewing the functional narrativization scale and behavioral mimicking and behavioral metaphor design strategies.

To help my discussion, I use the following diagram (Figure 17) adapted from Tanenbaum and Bizzocchi (2009) to map the Rock Band controllers and the Oculus VR interface onto an axis of functional narrativization from *literal* to *metaphorical* and *abstract*.

---

[5] Bizzocchi et al. (2011) define game metrics as "any mechanism used by the game to provide feedback to the player about the state of the gameplay or her performance within the game" in their discussion of narrativized game metrics.

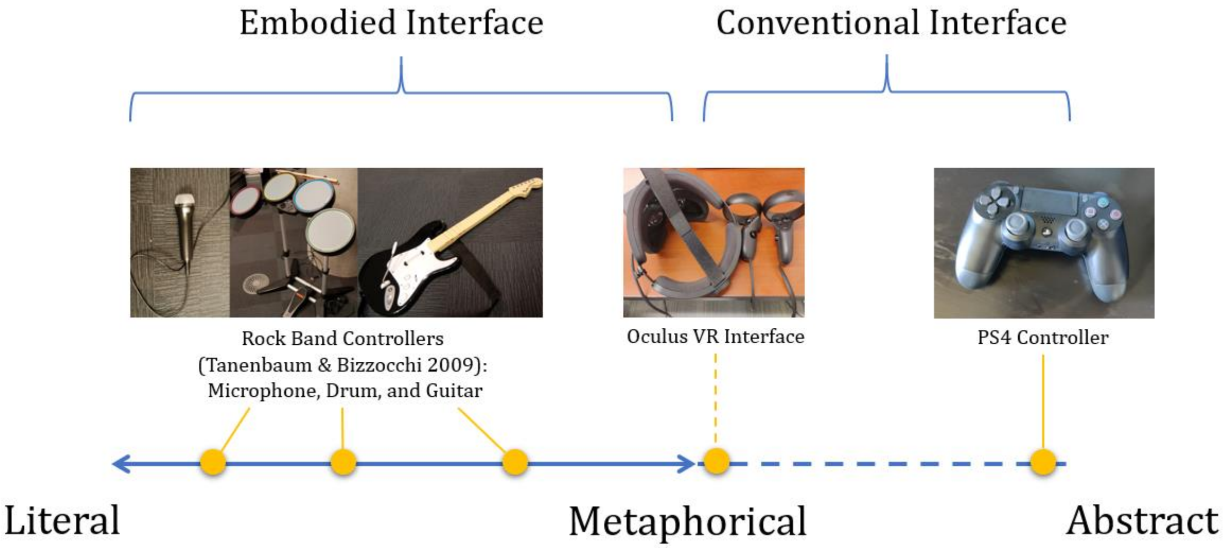


Figure 17. Metaphorical & Literal Functional Narrativization adapted from (Tanenbaum & Bizzocchi, 2009)[6]

Along this axis, the functional narrativization within the microphone controller is the most *literal*, as game interactions with the microphone *reproduce* the real-world referent. On the other hand, the drum and guitar controllers do not directly replicate how performers play them in the real world. Instead, they *evoke* real-world use cases. Therefore, in terms of functional narrativization, they are more *metaphorical*. On the category of the conventional interface, the PS4 controller affords little narrativized interaction and thus belongs to the *abstract* end.

The Oculus VR interface that I add to this axis shows a potential middle ground where a *conventional* game interface can support narrativized and embodied interaction. This is primarily due to the VR spatial tracking feature that affords bodily actions. As noted earlier, the metaphor of the VR motion controllers can be the player's virtual hands and what they hold or do in VR. Furthermore, with the *headset* also being tracked in six degrees of freedom (6DoF), the whole VR rig simulates a virtual head and hands that drive the player character's body in most first-person VR games. However, whether the VR interface is more metaphorical or abstract can still depend on the specific game interaction design and control mapping.

---

[6] Again, the position of the VR interface on this scale is subjective to the context of interaction.

Behavioral mimicking and behavioral metaphor are a pair of design strategies that further utilize the "Literal – Metaphorical" scale:

> **Behavioral mimicking** "deals with interactions in which the interactions required of the player by the game directly and *literally* mirror their real-world counterparts." "Behavioral mimicking merges the physical understanding of real world activities with the mechanics of the game play. In this strategy, the physical behaviors of the player's gameplay interactions mimic those of real life actions."
>
> **Behavioral metaphor** "deals with interactions that figuratively 'stand-in' for their real-world counterparts." "An interface can suggest a connection to real-world behavior rather than completely mimic it. This creates a *metaphor* rather than an identity with existing behavioral patterns." (Bizzocchi et al., 2011)

A closer review of the examples that Lin (2007) and Bizzocchi et al. (2011) use to identify the two strategies reveals a crucial factor - whether the interface *hardware* has enough DoF or natural inputs to afford a high degree of physically embodied interaction - gives rise to the difference between "mimicking" and "metaphor". The example interfaces of behavioral mimicking range from the Nintendo Wii gaming console to arcade dancing and driving game interfaces. Their games fall into the "physical interactive" genre using gesture, speech, text, vision, or other natural inputs that enable the design of behavioral mimicking. In contrast, the examples discussed for behavioral metaphor are mostly *graphical* interfaces on conventional platforms, such as the narrativized cursor of the mouse.

The VR interface hardware can be viewed as a fusion of an embodied interface and a conventional game controller. Although VR game interactions commonly use behavioral mimicking, whether a VR game interaction *should* follow behavioral mimicking is a design choice that is often influenced by the competing value of efficiency and game contexts. Whether it *can* use behavioral mimicking also depends on the technical constraints.

Bizzocchi et al. (2011) point out that the difference between behavorial mimicking and behavorial metaphor is the mimicking *fidelity* of the real-world behavior. The following section reviews the Framework for Interaction Fidelity Analysis that provides a systematic measurement of interaction fidelity.

2.1.2. Framework for Interaction Fidelity Analysis (FIFA)

McMahan defines interaction fidelity (IF) as the "objective degree of exactness with which real-world interactions can be reproduced" and proposes the framework for interaction fidelity analysis (FIFA) to determine the level of IF (McMahan, 2011). In their later review, McMahan et al. (2016) refine the framework to have the following categories of components: (for clarity, I bold the key criteria)

> "*Biomechanical symmetry* is the objective degree of exactness with which real-world **body movements** for a task are reproduced during interaction. It consists of three subcomponents:
>
> > *anthropometric symmetry* is the objective degree of exactness with which **body segments** involved in a real-world task are required by an interaction technique.
> >
> > *kinematic symmetry* is the objective degree of exactness with which a **body motion** for a real-world task is reproduced during an interaction technique.
> >
> > *kinetic symmetry* is the objective degree of exactness with which the **forces** involved in a real-world action are reproduced during an interaction technique.
>
> *Input veracity* is the objective degree of exactness with which the input devices **capture and measure** the user's actions.
>
> *Control symmetry* is the objective degree of exactness with which **control** in a real-world task is provided by an interaction. Control primarily concerns how the user's actions translate into system effects. It depends on only one component—*transfer function*[7] *symmetry*, which is the objective degree of exactness with which a real-world **transfer function** is reproduced through interaction.
>
> *Three Interaction Phases* (1) the initiation phase, (2) the continuation phase, and (3) the termination phase: The initiation phase includes all of the biomechanical, input, and control aspects required to begin using a technique. The continuation phase encompasses all of the aspects involved with continuing the interaction. Finally, the termination phase indicates what is required to stop the interaction." (McMahan et al., 2016)

Compared with the frameworks in narrative and embodied interface, FIFA is a step further to provide *objective* measures of the degree that game interactions *reproduce* real-

---

[7] The transfer function interprets and transforms input data into an output effect. In the original version of FIFA, besides the subcomponent of transfer function symmetry, the control symmetry component also includes dimensional symmetry and termination symmetry (McMahan, 2011).

world counterparts. However, it can *problematize* the concept and the area of game design when connecting IF to player experience and making design guidelines based on levels of IF.

By applying this framework to analyze interaction fidelity (IF) levels among related studies and connecting IF to its effect on player experience, McMahan et al. theorize IF as an *uncanny valley*[8] of VR interactions (McMahan et al., 2016). The high and low levels of IF lead to better performance and enjoyment, while the moderate levels of IF are the worst. The cases they reviewed include studies on *manipulation* (Zhai & Milgram, 1993; Mine et al., 1997; Ware & Rose, 1999), *navigation* (McMahan, 2011; Nabiyouni et al., 2015), *steering* (McMahan et al., 2010), and *search* (Pausch et al., 1997; Pal et al., 2016). McMahan et al. (2016) hypothesize the uncanny valley model of IF is due to the lack of *familiarity* in middle level IF and offer the design guideline of avoiding developing middle-level IF. However, it is worth noting that McMahan et al. themselves acknowledge that this theory should not be overgeneralized, and sometimes moderate IF can produce as good effects as high and low IF.

2.1.2.1. The problems with FIFA and the uncanny valley model

While FIFA is helpful to analyze interaction fidelity systematically, it has several flaws when applying it to the design space of modern VR games.

First of all, the "moderate levels are the worst" is simply too reductive to reflect the complex nature of game design. The fundamental flaw under this reductive claim is that it is based on lab studies that reductively link IF of VR prototypes to metrics of experience with little grounding in the rich *contexts* of modern VR video games. Although FIFA is a systematic way to describe IF, many more factors of both the game and the platform can still affect the *design choice* of the IF level. Contrary to the claim, the most commercially successful VR platforms and games today use a *mix* of IF levels, let alone many interactions they have developed span the medium level of IF. This point will also be echoed by my case studies of VR game interactions later.

---

[8] The term uncanny valley oringinates in robotics, it describes the relationship between the fidelity of a human-like object and the emotional response it evokes (Mori et al., 2012). The low end (e.g. industrial robot) and high end (e.g. healthy person) of human-likeness have higher acceptance while the middle level (close to the high level) of human-likeness (e.g. zombie and corpse) can be uncomfortable to human.

A recent study exploring IF within a modern VR platform suggests that the preferred level of IF varies from different kinds of tasks: for object manipulation tasks, high IF is preferred; for whole-body movements, moderate IF is desired (Rogers et al., 2019). On moderate IF, the authors contend that "VR games should not strive for full realism and instead offer more abstractions for increased usability and ease" and "explore the uses of substitutions and approximations of physical and cognitive challenges" (2019). These arguments are a starting point in revealing the duality and trade-offs between the values of simulation and efficiency, but their study is limited in scope. Still, it shows the fallacy of the uncanny valley model.

Also, FIFA only addresses the "action" part of interaction - or an interaction *technique*. However, it misses other related game aspects that can be part of the interaction *process*, such as signifiers and feedback.

For example, in a study on interaction and display fidelity on an FPS aiming task, McMahan et al. (2012) report that the effects of the fidelity components are nuanced because users always position an *always-visible crosshair* on targets to fire. However, I argue that the aiming crosshair is a factor of IF not included in the framework. In some VR first-person games, there is no such crosshair. Instead, they simulate the realism of mechanical aiming, such as the archery in my experience with The Elder Scrolls: Skyrim VR. In other VR FPS games such as Population One, the aiming crosshair provides rich contextual feedback to enhance the player's performance.

Moreover, the levels themselves overemphasized by the absolute and discrete (high, middle, low) values can be problematic. It is unclear what the thresholds are between the levels, though this can be a rhetorical issue. Moreover, the IF level has a high coupling with the *platform* that I think is a *variable* in different comparison cases. Also, its judgment call can be *subjective* depending on the weight assigned to each component.

For example, if we rule out input capacity for modern VR platforms, will biomechanic symmetry weigh more significantly over control symmetry in determining the level? Although McMahan (2011) suggests that control symmetry has the most significant impact, the result may still vary when the platform changes because the platform would determine the control and biomechanic components. Rogers et al. (2019) suggest that designers can explore cases of game interactions with more specific contexts by focusing on one modern VR platform, where the middle-level IF can be optimized.

From a design perspective, FIFA overly complicates the value of simulation and requires extra mental labor to determine IF; it ignores the equally important value of efficiency for all levels of IF. Although game interaction design can benefit from general interaction design principles, it rarely follows any "golden rule" solely based on fidelity levels. Furthermore, the components in FIFA are named with a lot of jargon, potentially limiting its popularity among designers. The expertise in game design is developed through practices, and how well an interaction works in the game is usually empirical, with particular reasons in the gameplay. While it is helpful to know the roots that affect IF as laid out in FIFA, it can be tedious and dogmatic to look at the detailed components whenever designing a game interaction. This is a reason why I want to propose a framework that consists of only two essential yet straightforward dimensions – simulation and efficiency.

Regardless of the problems of FIFA and the uncanny valley model of VR interactions, I still consider IF and particularly its biomechanical and control components as useful lenses to analyze the design value of simulation. However, what I believe needs adaption when applying this theory to modern VR games is the absolute and reductive views of the levels and the uncanny valley model that claims "moderate levels of IF are the worst." McMahan and the co-authors have involved performance evaluations of IF and approached performance in terms of efficiency (McMahan et al., 2012, 2016). Still, in the end, they overemphasize the role of fidelity, or in my words - simulation, but fail to evaluate efficiency within an interaction. As a result, their claims such as "high fidelity increased engagement and usability" may not always hold in VR game interactions.

2.1.3. Duality and Two-Dimensional Framework

So far, I have reviewed the two foundational theories that frame the design values of simulation and efficiency of game interfaces and interaction. Although they come from two different areas of study, they both address simulation *fidelity*, providing practical measures for it. Yet, they still have limitations. They tend to emphasize more on simulation and view the two values separately. As an extension, my study explores the *duality* of simulation and efficiency, which I argue are both indispensable and of equal importance for VR game interaction. I propose the following two-dimensional framework as visualized in Figure 18:

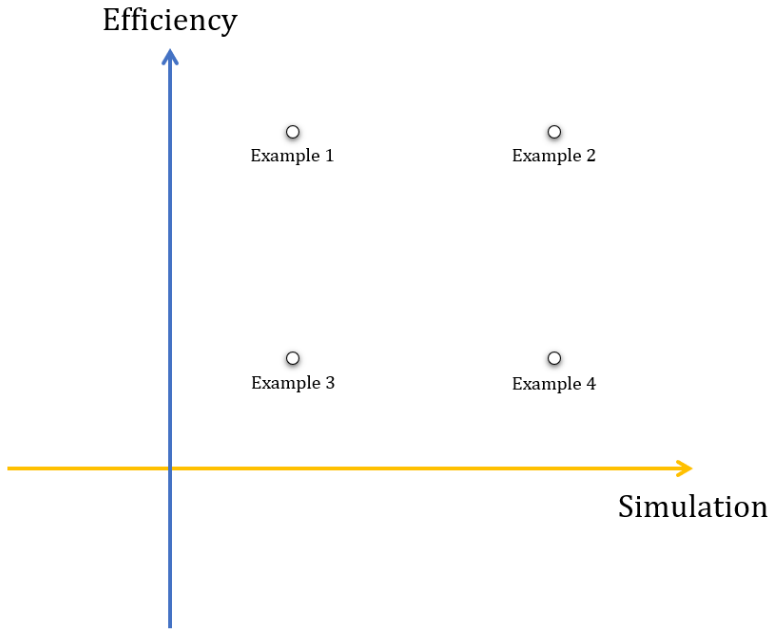


Figure 18. The simulation and efficiency framework

By mapping examples of VR game interactions onto this two-dimensional coordinate frame rather than separate continuums, we can more clearly see how successfully each interaction design achieves the two values. In addition, we can further explore the trade-offs and tensions between the two values for each example.

Besides the conceptual guidance from value tensions, my intention to frame the duality also comes from modern physics inspirations. For example, by considering the case of light-speed motion, Einstein's theory of relativity shows the duality nature of time and space that were thought to be absolute and unrelated in classical theories. By looking closely at the atom scale, the quantum theory reveals many new phenomena, including the quantum entanglement of particles and the fact that a particle's velocity and position cannot be both measured precisely and simultaneously. Furthermore, physicists seek to unify the two theories that apply to the macro-scale and the micro-scale. These theories show us that things we commonly view as separate concepts can be related when examining them in a particular domain.

VR is the particular domain of interaction design, like the macro and micro worlds are to the two outstanding theories of modern physics. Although studying the design values may not be as significant as modelling the physical world, the same logic of duality may still apply to produce design knowledge for new media platforms.

The two-dimensional framework of simulation and efficiency is a natural extension of the theories reviewed in this section by combining the two values that can correlate. Tanenbaum and Bizzocchi (2009) point out that the three vectors – ludic, kinesthetic, and narrative are not truly isolated from each other and can connect in broader themes such as immersion (Ermi & Mäyrä, 2005) and transparency (Bardram & Bertelsen, 1995; Bolter & Grusin, 1999). The ludic efficiency scale (problematic – simplistic) can correlate with the functional narrativization scale (literal – metaphorical – abstract) in specific interaction cases. For example, the PlayStation controller is generally considered as an abstract and problematic (inefficient) interface. Still, it may become towards the simplistic side of ludic efficiency when the interaction is button presses. The microphone controller is a literal yet highly simplistic interface, a rare example of achieving both simulation and efficiency. The guitar controller in *Rock Band* was viewed as having a moderate level on both scales. It could be that the simulation of the guitar controller does not perfectly connect to the prior knowledge of a real-world guitar, and therefore, the interface problematizes the gameplay.

Arsenault identifies another "half-simulated" guitar controller in the game Guitar Hero, where the guitar does not have strings, frets, or sound but accurately simulates melody, harmony, and rhythm that a real guitar can handle (Arsenault, 2008). He concludes that a simulation can work either in the direction of breath "by modelling as many facets of an experience as possible" or in the direction of depth "by simulating as closely as possible a few parameters of an object or experience" (2008). This also suggests that the full degree of simulation is not all that matters. It needs to co-function with efficiency and adjust its level with particular game interaction contexts.

Efficiency is also obviously not the single independent factor towards a pleasurable game experience. McMahan et al. attribute the efficiency of low interaction fidelity to *familiarity*. However, as Tanenbaum and Bizzocchi (2009) suggest, familiarity can problematize the use of a moderate-fidelity game interface, while mastering an inefficient game interface can be a source of pleasure. They contend that the efficiency in the interface needs to be balanced with the challenge to achieve the optimal flow state.

Therefore, the duality of simulation and efficiency is a natural extension beyond emphasizing them as separate values. We can understand how they balance and constrain each other under various circumstances of game interaction design.

The following section reviews broader VR interaction studies on interaction techniques. In addition, I introduce related game design aspects that I find useful to describe and decompose game interactions into formal elements. Finally, I contend that to evaluate the two design values within the interaction techniques, we need to connect them to the game context.

2.2. VR Interaction Techniques and Related Design Aspects

First of all, when I say "interaction", what exactly do I mean? Of course, it includes the action, which is the core part of any game interaction, but I think it should also cover the course of interaction and the game context. So when we talk about a game interaction, we should also consider the input, the feedback, related game aspects and the player's experience.

Caroux et al. define player-video game interactions as "interactions in which technical aspects of video games have influence on the players' engagement and enjoyment" (Caroux et al., 2015). As shown in the diagram in Figure 20, they propose the following framework to map related aspects of player-video game interactions.

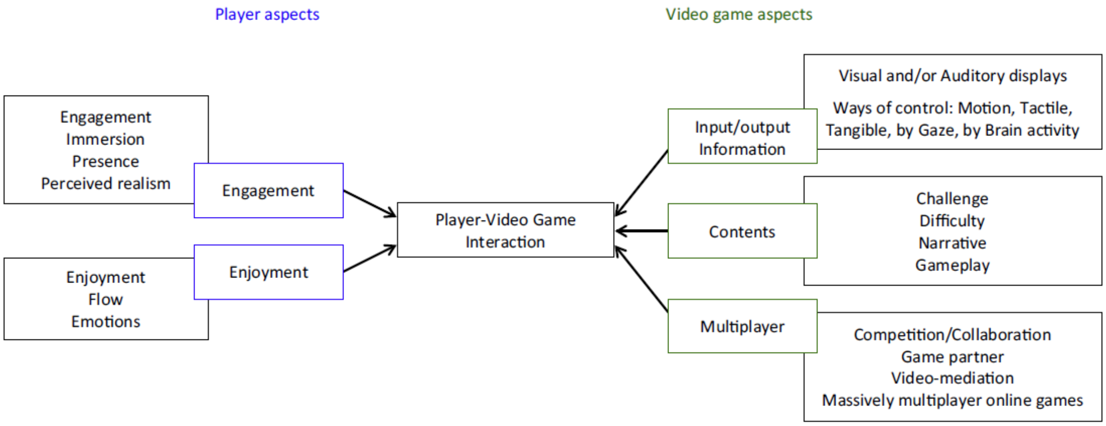


Figure 19. Player-game interactions and their related aspects in Caroux et al. (2015) 's review

Many of the existing VR interaction studies that we will review soon have focused on exploring the impact of an input variable on the player aspects. These studies mostly use precursors of commercial VR systems but have laid foundational work for modern VR interactions to follow and extend.

One widely studied area of VR interaction is locomotion techniques (LTs), which deal with how players move and travel in VR. Recent reviews and surveys on virtual LTs, such as (Di Luca et al., 2021; Al Zayer et al., 2020; Tanenbaum et al., 2020; Boletsis & Cedergren, 2019; Boletsis, 2017; Jerald, 2015), to name a few, have already covered hundreds of studies and examples of LTs, their taxonomies and comparison. Therefore, my point here is not to list and discuss those LTs exhaustively but rather to connect the most prevalent LTs identified by these authors to VR game contexts and my simulation-efficiency framework.

2.2.1. Semantics and Techniques of Game Action

Four types of VR LTs that these authors generally agree on are what Boletsis (2017) describes as follows: 1) *motion-based*, 2) *room-scale based*, 3) *controller-based*, and 4) *teleportation-based*. However, I want to discuss LTs beyond techniques to emphasize the meaningful *semantics* of the actions in the game. For example, to move the character, players can choose to push the joystick in a direction for continuous movement or use teleportation as the alternative. Here "move" is the *semantics* of the action in the game context, and the ways of operating the controller to either continuously move in one direction, or teleport to the selected location, are the *techniques*. Similarly, "climb" is the semantics of the action, and alternating movement of each motion controller back and forth in a specific direction is the technique. In the game, players and designers care more about semantics, and techniques are what players need to learn about the gaming platform and control. It is worth noting that sometimes one semantic action can have multiple implementations of techniques for players to choose from or switch between in the gameplay. I use the following diagram to conceptualize the relationship between techniques and semantics, which I find is a pair of useful vocabulary to describe game actions (Figure 20).

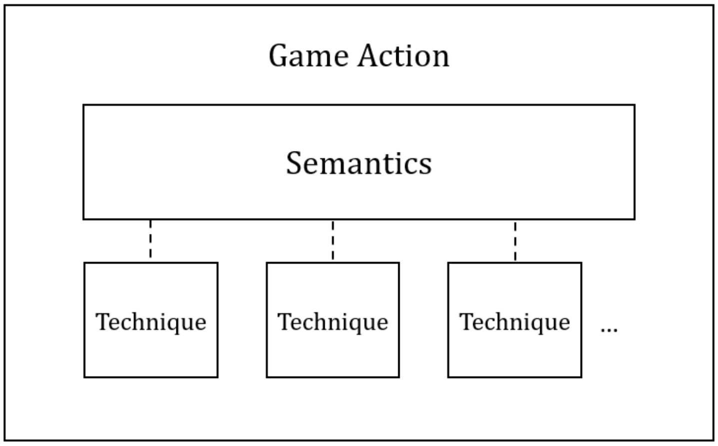


Figure 20. Techniques and Semantics of a Game Action

Techniques are simulations of semantics because they are the implementations of the game action. Techniques build on affordance and constraints at the gaming platform level. On the other hand, semantics describe a task or goal at the game design level. I will show the example of the input system in the Unity engine when I review platform studies in the next section. By conceptually decomposing a game action into these two levels, we can analyze how techniques simulate a game action described by the semantics.

Among the four types of LTs identified in Boletsis' work (2017), *motion-based* and *room-scale based* techniques both have more physical movements than *controller-based* and *teleportation-based* techniques. Examples of motion-based LTs include walk-in-place, redirected walking (which use techniques to reorient players in the physical space to walk continuously in VR), and arm swinging. Examples of room-scale based LTs include real-walking, which supports players to walk freely in physical space until they reach the boundary. The difference between motion-based and room-scale based is that motion-based supports continuous movement in open spaces while room-scale based is limited by the size of the environment (or a room).

Most commercial games do not support motion-based LTs because most players do not have a larger open space than their rooms at home. Real walking, controller-based, and teleportation-based LTs are ubiquitously supported in modern VR games. Some games can even enable them at the same time.

In the VR LT literature, the real walking technique is often considered superior for moving and turning in various tasks (Al Zayer et al., 2020). Real walking has a high interaction fidelity due to its biomechanical and control symmetry. However, real walking is often not the primary LT in modern VR games because it is not viable to travel large virtual spaces. In contrast, players typically use a mix of controller-based and teleportation-based techniques for movement and turning, such as the examples in Skyrim VR that I describe at the beginning of this dissertation. Real walking can be useful to make minor adjustments or navigate a room-scale VR environment. However, real walking can be further limited by the "seated play" style chosen by many players to avoid the fatigue in the standing style.

Technically, Boletsis' controller-based and teleportation-based can be further unified into controller-based because teleportation is typically enacted by pointing the motion controller and pushing the joystick or pressing the buttons. Even room-scale based LTs such as real walking can be controller-based if we consider the head-mounted display as a *controller* of the head position and orientation in VR. Therefore, I do not find the overall types of LTs from Boletsis and related studies useful for analysing VR game actions. They generally focus on techniques and their performance in tasks but do not connect to gaming contexts.

Semantics is a more useful description of game actions because it opens up many more meaningful actions than academia's focus on basic LTs. For example, besides move and turn, the locomotion techniques for climbing, sprinting, crunching, jumping, flying, sliding, dashing, dodging, swimming (to name a few) and their performance in the game context are underexplored. When the number of locomotion and other game actions increases, more design problems and tensions also arise.

2.2.2. Control Scheme and Action System

One of these design problems is the control scheme, which connects physical inputs on the controller to interaction techniques and deals with the assignment of the inputs to multiple game actions. The control scheme of modern VR interfaces can influence aspects of player experience, such as presence, enjoyment, and flow, as demonstrated by related studies (Limperos et al., 2011; Martel & Muldner, 2017; Shelstad et al., 2017; Seibert & Shafer, 2018). However, on the point of control scheme, these existing studies are still far from what I want to explore in terms of scope and depth. They mostly compare a VR control and a non-VR control, or two variations of controls. They also only examined basic tasks such as locomotion and targeting. Still, I find their evaluations reflect the value of simulation in terms of the sense of being natural and realistic (Seibert & Shafer, 2018) and the value of efficiency in terms of accuracy and completeness (Martel & Muldner, 2017).

I argue that the control scheme is a much larger issue than existing studies have explored, and it scales with the number of actions in the game. It becomes a puzzle to solve when there are over twenty or thirty game actions and only ten available physical inputs. Some game genres, such as RPGs and FPS games, are naturally action-rich. As we will see in the analysis chapter, control schemes for these games become a place to reveal and resolve tensions between simulation and efficiency.

I use an action system as a larger unit to further study the control scheme of commercial VR game actions. There are two meanings of the word "system". First, an action system, such as a combat system and a shooting system, refers to a group of game actions that form a system representing broader meaning than its components. For example, a combat system may include swinging the sword, powerful attacks, and even locomotion interactions like dodging attacks and object interactions like consumables.

Second, I intend to systematically study a game action by including non-action components, such as controls, signifiers, and feedback. The term signifier is a useful concept in design. As Norman explained, it is the perceivable part of an object's affordance that functions as cues for people to cope and understand in this social and technological world, and thus designers should provide signifiers (Norman, 2008). From a linguistic point of view, a signifier refers to the concrete embodiment of the (signified) message, and the unit of the signified and the signifier together help us recognize and understand (Pearce, 2012; Barthes, 1977). The action is the central component, but the non-action components can reflect the design values within the interaction. The action can consist of *systematic* controls, components, and steps.

Taking these perspectives together, I use the following diagram to illustrate the concept of an action system.

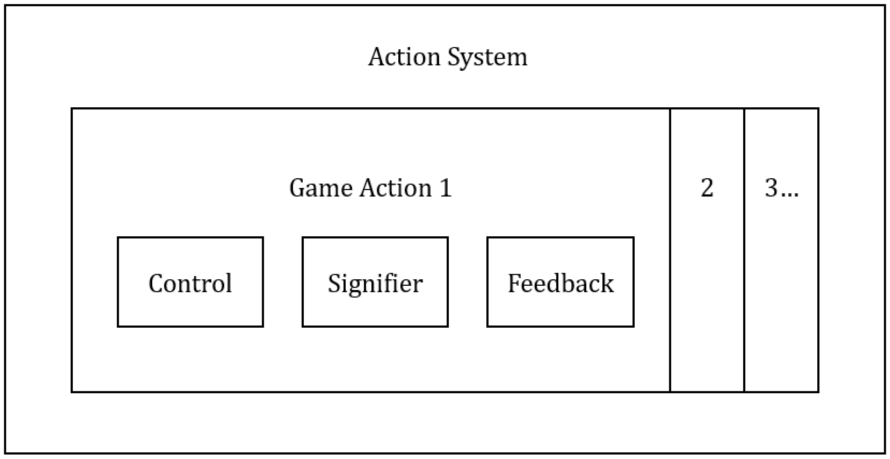


Figure 21. Action System as a unit of study involving multiple game actions and related components

I do not think the action system is a new invention because game actions are inherently systematic. They are what game developers use as design patterns to pass between their projects. My goal of pointing the systematic nature of game actions is to provide locations to examine the values of simulation and efficiency. However, when it comes to control schemes in VR, the action system becomes a more salient concept because both an action and an action system can require a series of controls, unlike simple keypresses in traditional platforms. The chapters I lay out for the analysis – locomotion, combat, and object interactions can co-function and involve each other in the gameplay and form more extensive systems and hierarchies. Therefore, when we talk about control schemes, we must clarify the scope of actions systems. While each action or action system builds on a control scheme, such as those discussed in the literature, the whole game calls for a control scheme solution that solves the entire puzzle.

Besides VR locomotion techniques, researchers have also explored a few other interesting game actions and their controls, signifiers and feedback. For example, gun shooting is a common FPS action system that involves multiple actions, including aiming, firing, reloading. Krompiec and Park propose a two-handed technique for gun shooting and an improved reloading design, as demonstrated in comparison with other methods in Figure 22.

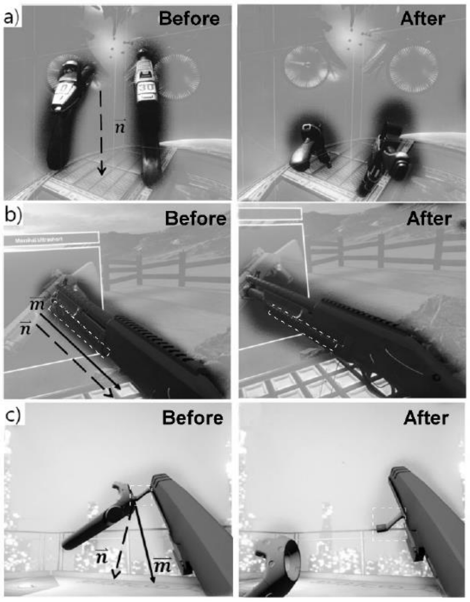


Figure 22. a) A simple reloading approach. b) A highly complex and realistic approach. c) The improved solution with more stability by projecting the physical motion onto the gun model (Krompiec and Park 2019)

By projecting the actual physical motion of reloading (vector n in the diagram) onto the direction of the gun model (vector m), this solution enhances stability while keeping the action realistic. The authors also present other interaction models, such as grabbing and attaching ammunition to the gun, reducing each action's execution time to close to one second. Besides control, they also evaluate different levels of haptic feedback to enhance the player experience with the gun. Their solutions reveal how the technical aspects of game actions, such as control and feedback, can balance the values of simulation and efficiency.

Haptic feedback is a new feature with modern controllers, and it can simulate various vibrations and textures in the game. Haptics in VR can be parameterized by frequency, amplitude, waveform, duration, and rhythm (Brewster & Brown, 2004). However, how these parameters can function together to enhance player interactions in VR is still unexplored. Besides haptics, visual cues are more common design elements for signifiers and feedback, and they apply classic design principles in VR. For example, a study shows that the visual cues of the temperature of a mug significantly affect how players grasp it in VR (Blaga et al., 2020).

So far, this section has paved the way to explore simulation and efficiency by locating several components and layers within a game interaction. The following section will further dive into the platform underlying the interaction and discuss how technical aspects can shape the design choices and values.

2.3. Platform Studies and My Previous Work on Extended Reality

Platform studies is another lens that I will use to study VR game interactions. Platform studies examines the underlying computing platform that supports the creative content and is first established in game studies in Montfort and Bogost's book on the Atari VCS gaming console (Montfort & Bogost, 2009). In their work, they investigate the technical capacities of the classic gaming console, such as the chips, the software code, and the graphic display. This platform perspective leads to the productive MIT book series on Platform Studies. Later books in this series, such as *Codename Revolution* (Jones & Thiruvathukal, 2012), *I Am Error* (Altice, 2015), and *Now the Chips Are Down* (Gazzard, 2016), have extended the study from the technical aspects to the broader culture and contexts within which the platform and content is published, taking user documentation, the lesser-known history, and social interactions into account.

As I frequently mentioned in previous discussions, the *technical constraints* of VR platforms can shape the interaction design of game actions. Particularly, by applying the platform studies lens, we can explore how aspects of the VR platform reveal, amplify, and resolve the potential tensions between simulation and efficiency. While many platform studies take a media-archaeological approach (Apperley & Parikka, 2018) to revisiting old platforms, my work in this sense serves the current movement of VR by capturing design knowledge from the state-of-the-art commercial VR platform and published games.

I present the following cases of extended reality (XR) to further show my interest and positions in adopting platform studies to study VR game interactions and reinforce the focus on the technical aspects of the platform. XR is an umbrella term for virtual, augmented, mixed, hybrid physical-digital realities and environments of human-computer interactions. One salient feature of XR that sets it different from traditional screen-based platforms is that XR uses spatial computing as its core mechanism, which enables game interactions to be more embodied.

The three cases are from my previous work on XR. They cover a wide range of aspects, from experience design in mixed reality, analysis of the computing and interaction models of real-life escape room games as an analog platform, and the software development of the XR interaction toolkit for the Unity game engine. I review each case from the platform perspective and connect them to this study. At the end of this section, I will discuss the Oculus Rift S platform, which is the platform of this study.

2.3.1. Case 1: Designing Mixed Reality Transformative Play

This game project, named *Magia Transformo: The Dance of Transformation*, is a design experiment to explore how tangibles, diegetic interfaces, and enabling technologies of mixed reality can build a transformative experience for co-located multiple players (Jing et al., 2017). The design process embodies principles in *transformative play* (Tanenbaum & Tanenbaum, 2015), which brings techniques from theater practice for training actors, such as backleading[9] and “outside->in” transformation[10] , to guide players’ activities and support their identification with characters.

2.3.1.1 Game design and setup

Magia Transformo sets up the following technological-mediated stage and the gameplay narrative to approach the design goal of transformation. The game space is a ritual chamber with a magic circle on the ground. A cauldron sits in the center of the magic circle, emitting chromatic LED lights and smokes to produce various signals. The ritual is for three players to unlock magic spells cooperatively by following instructions from their magical mentor, Alistair, to dress up as witches and warlocks and perform multiple cosplay activities. At the far side of the chamber, each player picks up a hat and a cloak, choosing from six pairs that each resembles a magic element: fire, water, air, earth, energy, and darkness. Each of these costumes has a magical medallion with a laser-cut alchemy symbol and an RFID tag inside. Near the costumes is a table on which a magic altar and three magic books await for players. The altar has an RFID reader and an Arduino microcontroller to communicate with the main game program on a PC. Each spellbook contains an android smartphone inside that also communicates with the main game program.

Figure 23. Magia Transformo interfaces: Cauldron, Altar, Spellbook, Costumes

---

[9] Backleading is a technique used to help untrained audience become successful participants and performers in interactive theatre experiences (Wirth, 1994). Many backleading techniques such as direct verbal instructions are like game mechanics such as progressive quests and tasks.

[10] Outside->in methods use external systems of enactment, such as scripts, body language, costumes, and props to help the actor produce the internal cognitive effects and transform into the character (Stanislavski, 1936; Benedetti, 1997; Krasner, 2000; Stanislavski, 2013).

Figure 24. An overview of the setup (at Indecade 2017 game festival)

The gameplay has two phases. In phase one, Alistar, the magic mentor, welcomes three initiates as they step into the chamber. Then, following his vocal instructions (backleading), players take turns to select and register a hat and a cloak on the magic altar before equipping them. Once a player places the magic medallion of a costume piece on the altar, the spellbook lights up, showing the character with the corresponding costume pieces and magical elements. At the same time, the cauldron turns into the corresponding color of the magical element, with a sound effect to signal the magic power that the player has picked up. Finally, when dressed up, the player holds the activated spellbook and steps onto the magic circle, waiting for other members to join. Players can click on the spellbook interface to explore the stories and the lore of the game world built upon the combinations of magic elements from the hats and cloaks.

The lore of the game is based on magical element combinations, which give birth to different schools of covens. The following diagram visualizes the core part of the game world.

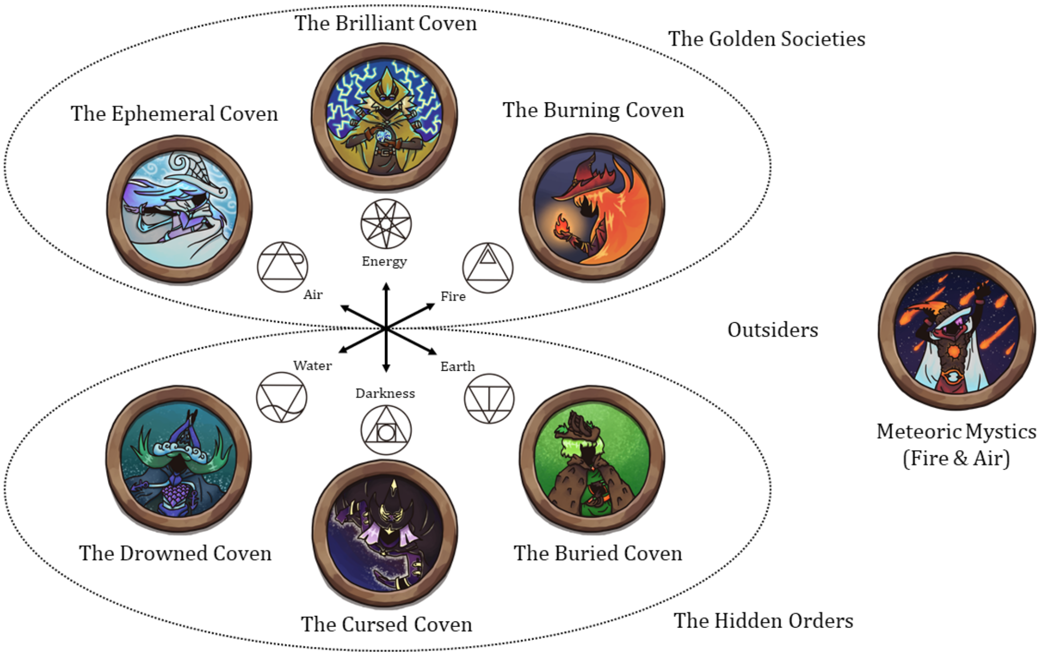


Figure 25. The lore design of MagiaTransformo based on magic elements

When all three players have gathered at the center, the game enters phase two, the dancing part of the ritual. Players follow directions on their spellbooks and voices of Alistar to perform a series of ritual activities, which include moving around the magic circle in the same direction together, spinning around themselves, and shouting out their verse displayed on the spellbook. In the end, players move to the center and raise their spellbooks, marking the climax and finishing of the ritual. The dancing music stops, and the cauldron flame goes crazy. With the sound of spellcasting, players accept the result of their practice. If they successfully unlock a spell, it will be added to their spellbook; if they fail, they can try a different combination of witches and the ritual again.

Figure 26. Phase two of the ritual. Players gather around the cauldron to begin the dancing part (left); the spellbook interface gives each player directions and tasks, such as move around the circle to their right (middle); players follow the directions and dance around the cauldron to summon spells (right).

2.3.1.2 Design challenges from the platform

This design case reveals how the platform can afford and constrain unique mixed-reality gameplay activities that are physical and embodied in their nature. We find that the complex technological systems underlying the game interfaces successfully hide from players' attention, making most players immerse in the activities and narrative of the gameplay. We observed quite a few moments where players are in character and enjoy the gameplay. However, there are also a few design challenges where the technical aspects confuse players and limit the scope of gameplay. To begin the discussion on the platform perspective, let us first take a look at the system architecture shown in the following diagram in Figure 27.

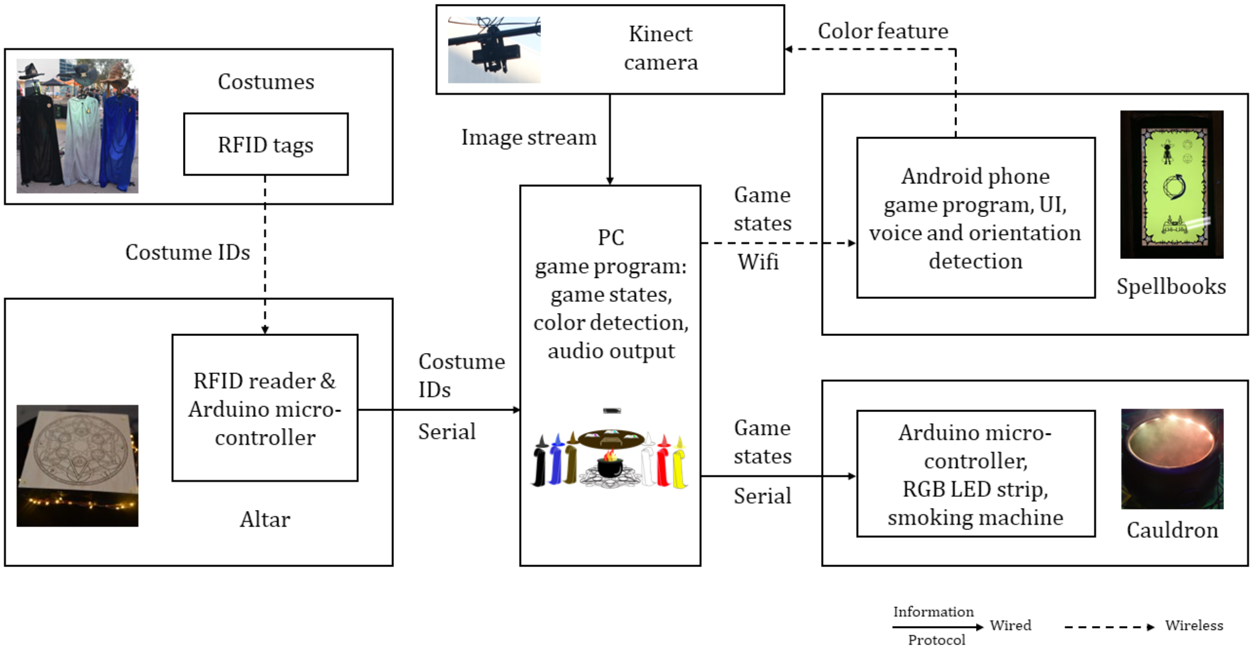


Figure 27. System block diagram of Magia Transformo's game interfaces and technologies

Although this platform is an ad-hoc solution using limited resources, many design challenges emerging from it correspond to the challenges of VR interaction.

The first challenge is that this game has minimal control inputs that have to support multiple complex interactions, which resemble the case of VR motion controllers. In phase one, the game system only knows what RFID tag was scanned, but not who scanned it and whether that person has worn the item. Therefore, we use the backleading method to guide players through the voice of Alistar directly. In phase two, for the dancing verification, the game uses the camera and image processing to track the position of the spellbook's screen based on its color feature and determine whether three players have rotated enough around the magic circle. This vision-based technique works in our lab settings, but in the live demo, we changed to a wizard of oz approach of simply counting several seconds after asking players to dance. To my surprise, all groups followed the instruction and actively danced around the circle. Some of them even performed exaggeratively in their characters. When they were well prepared in costumes, at that moment, hearing the dancing music and Alistar's voice, they just reacted with each other starting to rotate in the same direction, without questioning whether this part is a mandatory task of the game and how the system verifies their moves. This shows the power of backleading and "outside->in" transformation designs to compensate for the short of system verifications, not to mention the fact that the festival atmosphere and our delicate stage setup contribute to the experience. I think the design achieved both values despite that not all the technologies were able to.

Other implementations of verifying the ritual activities yield mixed results. The spinning detection, verified by logic states from the input of the orientation sensor in the smartphone, worked robustly in the live showcase. For the voice input for shouting out the verse displayed on the spellbook, we avoided machine learning-based approaches of recognizing what players say, considering the accuracy in the noisy environment and the computing power of the devices we had. Instead, we verified it based on a simple threshold detection of the microphone input volume. However, this part sometimes appeared to be glitchy in the show. We suspect that the input voice volume could depend on the player and be contaminated by noise. Even today, I still do not think there is a neat solution to verify the voice input in that situation.

Another design challenge we observed is shifting players' attention between the spellbook screen and the physical space. For example, at the beginning of the dancing phase, when players see the spellbook UI (Figure 26) indicating they need to move around the circle, many use their fingers to perform a rotation gesture on the screen before they realize they can and should move *physically*. This UX problem happens so often that we have to add a direct text instruction saying, “move to your right” on the screen. We hypothesize that players tend to take the more efficient and familiar action when multiple techniques connecting to the same semantics are available.

There are similar design cases in VR games where players need to maintain awareness of the 3D space while tunnelling their attention onto interfaces closer to their virtual bodies, such as the objects and UI on their hands. Due to the spatial interaction nature of these XR platforms, they commonly face the design challenges of navigating players’ attention to look at and do the right things that designers want them to look at and do. To overcome these challenges, designers can implement additional signifiers and cues to improve the efficiency of interactions.

If we view the interaction techniques that we implemented for this game as simulations of the ritual activities, the challenges of the platform aspects that we discussed above show the limitations of the simulations. The perfect simulation perhaps is not to implement any system verification that may have flaws but rather use “human judges” to confirm that the actions have been taken by the players and press the continue key. However, this is unrealistic. Whenever game designers employ digital technologies for automation and efficiency, the technologies themselves usually become the gap of the simulations. In Magia Transformo, we hide the technological components inside the diegetic interfaces, making the ugly computing parts invisible to create the magical experience, but this only resolves the visual aesthetics aspects of the gap. The system aspects of triggering and verifying embodied actions and the nature of spatial interaction may still be tough challenges for games like this. While the goal is to explore the design space of transformative play in mixed reality settings, we acknowledge these challenges that may constrain the scope of the gameplay and more in-depth experience, such as character identification.

The next case of escape rooms further highlights the roles of the materiality of the platform and human bodies in the computing and interaction of an analog platform.

2.3.2. Case 2: Computing Models of Escape Rooms

Escape rooms have become a popular form of party entertainment games in recent years. Escape rooms are live-action puzzle-solving adventures. A group of players typically start by entering a room with a narrative of being trapped, and they have to find clues and solve logic puzzles to escape within a time limit. Scott Nicholson, a pioneer scholar of this field, suggests that escape rooms are a convergence of several precursors: live-action role-playing, point-and-click adventures, puzzle and treasure hunts, interactive theatre and haunted houses, adventure game shows, and themed entertainment (Nicholson, 2015). He further proposes an “ask why” model for designers to create engaging escape room game experiences by logically and consistently matching various game design aspects: genre, setting, world, narrative, and challenges (searching, puzzles, tasks), critiquing abstract, random, and unreasonable designs that break the immersion of storytelling (Nicholson, 2016). Like the interface design in Magia Transformo, many escape rooms have high-quality diegetic props and seamlessly hide computing components inside them.

2.3.2.1. The platform and models

Game scholars have extended platform studies from digital platforms to analog platforms by investigating how analog components manifest computing behaviors. Their platform analysis starts at card games. Nathan Altice examines five characteristics of the playing cards that support computing and the generation of bits: (1) planar, (2) uniform, (3) ordinal, (4) spatial, and (5) textural (Altice, 2014). The planar form, for example, affords switching the card between facing up and down to represent one-bit information. Jan Švelch discusses how *Magic: The Gathering* employs these characteristics and how the materiality of the game as a platform is further shaped by community practices such as manufacturing and the secondary market (Švelch, 2016).

Besides the focus on materiality, the players themselves are considered as the source of analog computing (Bellomy, 2017). Bellomy radically amplifies the role of the human brain that assigns significance to the varying physical attributes in all analog games. He argues that even in digital machines, the outputs of transistors are subjective to how we count them. However, he acknowledges that the scale of the non-human components in those digital systems makes us believe that the system is doing the computing job, while in analog games, the function of people in computing becomes more salient.

Based on these two perspectives of materiality and people, I investigate how objects and minds function to generate computing in escape rooms (Jing, 2020). I seek a balance between the two views, contending that they both play indispensable roles in the game interaction.

First, I identify the one-bit components, such as the door with the two logic states of opening or closing and the key-lock puzzle with the two logic states of being solved or remaining unsolved in the game. The door's solid, untransparent material properties allow it to function as a gate to separate game scenes. Opening the door often means the meta puzzle of the game scene is solved. Although this means remarkable progress to players, in the sense of computing, it is just a one-bit update.

Then, I generalize the “one-bit processor” concept to *every* puzzle in the game and further establish models of designing and solving the puzzles. From the designer’s perspective, I use the *object graph* model to describe puzzles. From the players’ perspective, I use the inventory list and the puzzle list to model the dynamic process of searching for clues and solving puzzles. The following diagrams present the two models.

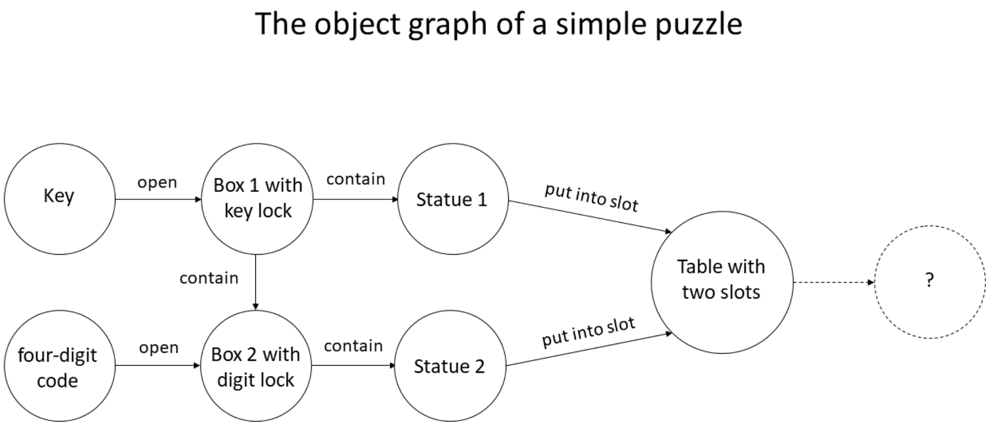


Figure 28. The designer model: the object graph of puzzles

This model uses the *graph*, a data structure in computing science, to represent objects as the notes and interactions as the directed edges between notes. This example imagines a simple puzzle that manifests as placing two statues into the two slots on a table. Each of the statues is gained from solving a box puzzle. The first box can be opened with a key. The second box is inside the first box and requires a four-digit code. The puzzle-solving process is illustrated in Figure 28, with the annotations on the edges indicating the interrelationship between objects.

This model provides an object-based definition of puzzles. A puzzle is a designed interaction between objects that manifest as a challenge to be solved. It is worth noting that puzzles can be hierarchical, as a large object graph can be decomposed into smaller sub-graphs, but they all share the same structure of clue objects pointing to the puzzle node. This is like the mathematical concept of *fractal* applied to puzzle design. Escape rooms designers can use this universal model in multiple ways, from visualizing the overall paths and connections of a network of objects, maintaining the technology needed for each object and a bill of materials, to assisting players in the game based on the states of the objects.

Unlike designers, players only see parts of the object graph rather than the whole picture when searching for clues and solving puzzles. Their gameplay behavior engaging with the clue and puzzle objects can be modelled as a loop of updating an inventory list and a puzzle list (Figure 29).

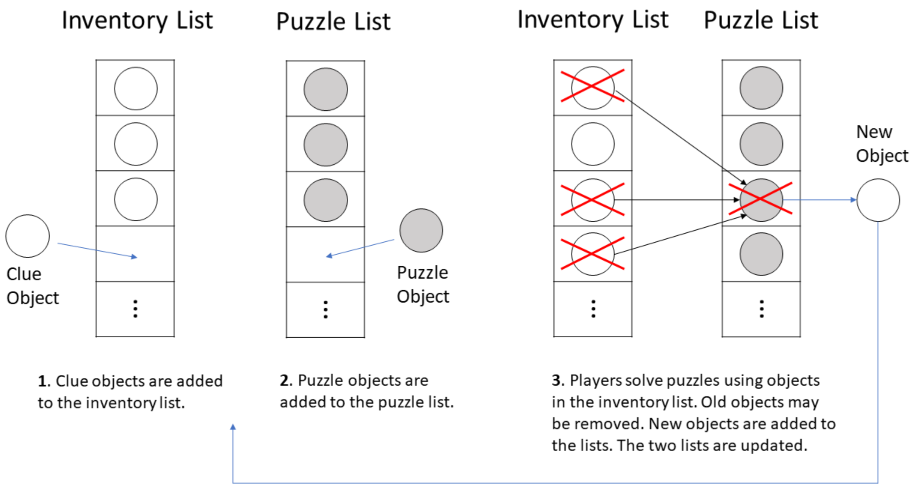


Figure 29. The player model: the loop of updating the inventory list and the puzzle list.

As an overview that combines the designer and player perspectives, the following diagram in Figure 30 shows the computing process of the object graph.

2.3.2.2. Materiality and people

Rather than repeat the explanation and discussions I have made in the original paper, I want to re-focus more on the two aspects of the platform, materiality and people, that I intend to highlight from this case and further correspond to VR.

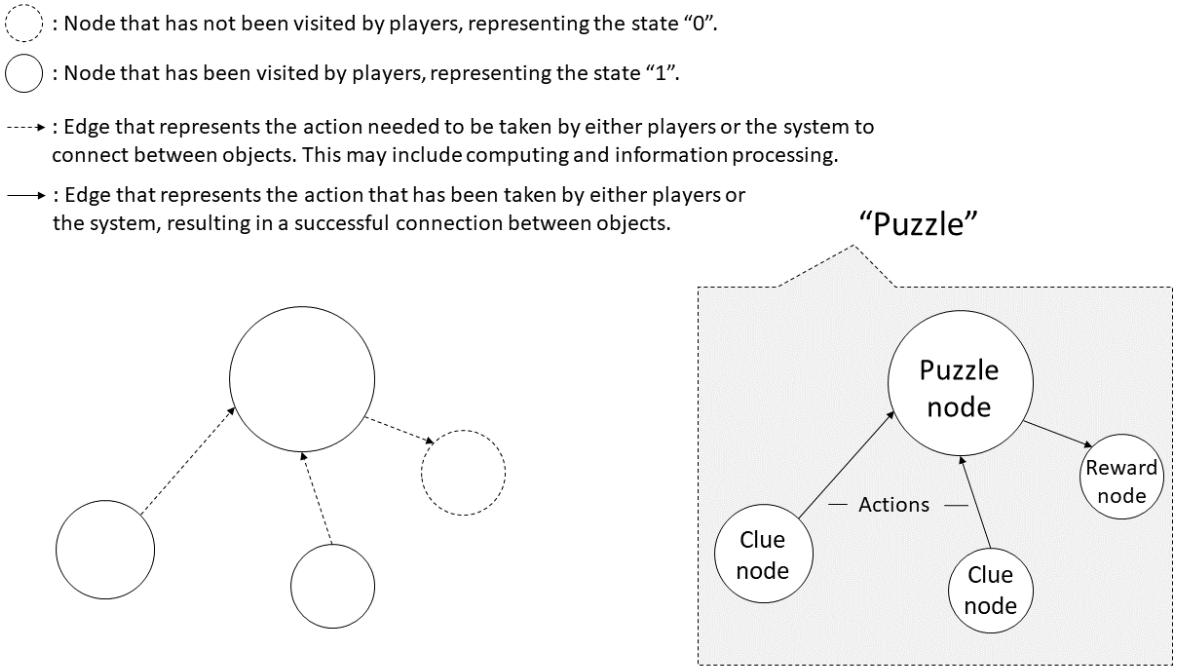


Figure 30. An overview of the object graph model. Left: players discover part of the nodes (which can be a clue or a puzzle node) and have not figured out the connections. Right: players have discovered all the nodes that form a puzzle and taken the right actions to figure out their logic connections. The reward node of the puzzle is unlocked.

The object-based models show how the materiality of the game platform affords players' computing and interaction at the macro scale. The art of design lies not only in creating the nodes, but also in the connecting edges that I acknowledge the models are an abstraction of the rich space of puzzle design. There are more specific manifestations of how the connections between nodes involve various player actions and their communications during the gameplay. Still, they all exemplify the cofunction of materiality and people in the system's operation.

Furthermore, in some escape room games, even players' bodies are used as materials to solve puzzles. For example, I once experienced a puzzle that required players to link their arms and hands together to conduct an electric current between two objects at the opposite sides of a room. Another example is the design where a player has to step into a cabin to use the body's weight to trigger the puzzle. In smaller areas, only a limited number of player bodies can be accommodated. Therefore, the interaction dynamic and the computer power of players are also constrained.

While VR is different from the escape room medium, the platform perspectives of materiality and people's bodies in constrained physical space must also be considered when designing VR interactions. The materiality of the VR platform is highlighted by the motion controllers and the non-transparent but immersive head-mounted display (HMD). However, the physical setting of the play space should also not be ignored. As I mentioned earlier, the motion controllers already have limited inputs, and the locations of these inputs further constrain their functions. In addition, the opaque HMD isolates players from the physical environment, further depriving their confidence of freely moving and turning around. As a result, many VR players use seated mode rather than room-scale and use controller-based locomotion techniques that most games have implemented.

These materiality and body perspectives are salient in VR, and they may bring additional design challenges and tradeoffs in interaction design. For example, the seated mode may lead to higher discomfort when experiencing fast movement interactions, like dodging incoming attacks. In addition, if sitting close to the computer desk, there will be potential to hit the desk with the arm when swinging the weapon. Also, the player's mind may lose direction in the physical world after occasionally performing physical turning. Therefore, these platform constraints may affect the design choices between the values of simulation and efficiency, and we will further examine specific design cases in later analysis.

2.3.3. Case 3: XR Interaction Toolkit

The last case of my previous work shows some *software* perspectives of the VR platform by reviewing some developmental and technical details of the Unity engine's XR Interaction Toolkit (XRIT).

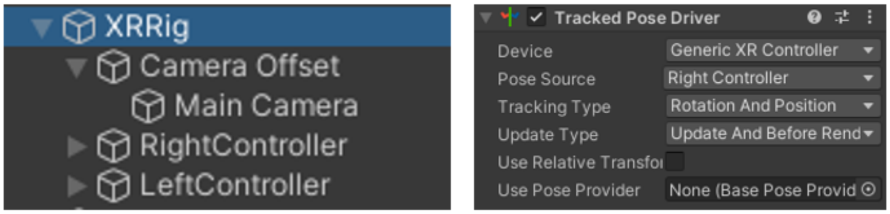
To give some background, I worked as a software engineer for XRIT at Unity Technologies for a total of six months. Many VR games and applications are made with Unity. The toolkit is an optional package that aims to facilitate the development process by providing solutions to hardware integration and basic interactions, so that developers can work on more meaningful gameplay programs without building the wheels. XRIT has already been released when I write this dissertation, and its code and documentation are all publicly visible. Therefore, I can reflect and comment on some features and their technical details without offending the company.

2.3.3.1. A common ground for XR platforms and interaction

When game developers start an XR project, they need to choose which hardware platform they develop with and deploy their game on, and there are a few brands of commercial VR headsets available in the market, such as Oculus, HTC Vive, Valve Index, PSVR, to name a few, and mobile or wearable AR and MR platforms. However, having to set up and manage each platform separately can be cumbersome in the long term.

Following the "unity" philosophy of supporting all major platforms and easing the development process, the XRIT provides a unified solution to integrating all kinds of XR devices and implementing common basic interactions with them. This common ground is based on the fact that although they come from different manufacturers, they can follow standard protocols, such as OpenXR, to translate their inputs to software code and apply universal software code architectures to organize their components and hierarchy. Also, most games will have to implement a set of interactions as their starting points, such as movement, turn, and grabbing objects.

For example, for VR and wearable AR or MR headsets and controllers, a common way to simulate them as game objects is the *rig*, which contains a camera representing the headset and two hand controllers. Both the camera and the two controllers support 6DoF tracking, meaning that their positions and orientations can be used as inputs for game activities. By adding the “XR rig” and its associated components to the scene, developers can view the virtual environment in three-sixty degrees and see their virtual hands in the game scene.



☑ Tracked Pose Driver
Device: Generic XR Controller
Pose Source: Right Controller
Tracking Type: Rotation And Position
Update Type: Update And Before Rend
Use Relative Transfor ☐
Use Pose Provider: None (Base Pose Provid

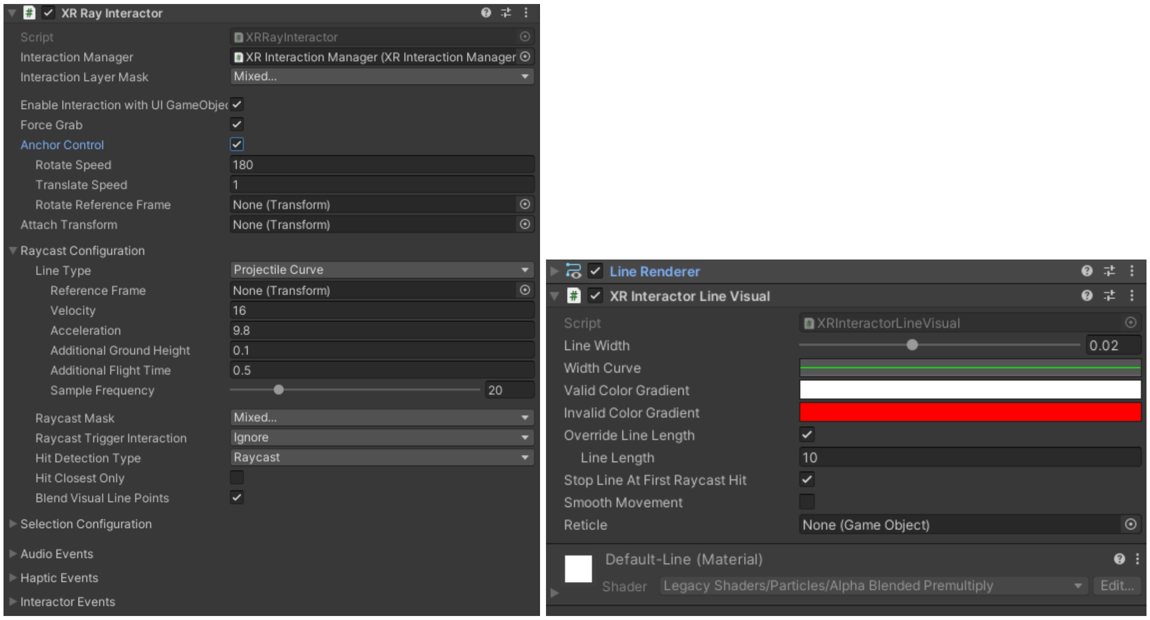
Figure 31. The XR rig is a universal hierarchy representing an XR platform with a headset and two controllers (left). A component attached to the camera and two controllers to support 6DoF tracking (right).

Once the “hello world” of viewing in VR is completed, users may want to add more interactions in VR. On a high level, XRIT provides a set of interactors and interactables (objects) for implementing basic interactions. The interactors are named after their control functions, including the direct interactor, the ray interactor, and the socket interactor. The interactables are named by their affordances, such as grab interactable and teleportation interactable. Their roles and some detailed technical options are further exemplified in the locomotion and object interactions that XRIT provides.

2.3.3.2. Ray interactor, teleportation, and grabbing objects

The ray interactor is a commonly used component by the controller. It uses the raycasting mechanism to support interactions from a distance, such as teleportation and grabbing a remote object.

Even the concept of ray interaction is simple, the number of functional options associated can be quite large. Here I want to point out a couple of things related to the later analysis of VR game interactions.



XR Ray Interactor

Script	XRRayInteractor
Interaction Manager	XR Interaction Manager (XR Interaction Manager
Interaction Layer Mask	Mixed...
Enable Interaction with UI GameObjec	☑
Force Grab	☑
Anchor Control	☑
Rotate Speed	180
Translate Speed	1
Rotate Reference Frame	None (Transform)
Attach Transform	None (Transform)
Raycast Configuration	
Line Type	Projectile Curve
Reference Frame	None (Transform)
Velocity	16
Acceleration	9.8
Additional Ground Height	0.1
Additional Flight Time	0.5
Sample Frequency	20
Raycast Mask	Mixed...
Raycast Trigger Interaction	Ignore
Hit Detection Type	Raycast
Hit Closest Only	☐
Blend Visual Line Points	☑
Selection Configuration	
Audio Events	
Haptic Events	
Interactor Events	

Line Renderer

XR Interactor Line Visual

Script	XRInteractorLineVisual
Line Width	0.02
Width Curve	
Valid Color Gradient	
Invalid Color Gradient	🟥
Override Line Length	☑
Line Length	10
Stop Line At First Raycast Hit	☑
Smooth Movement	☐
Reticle	None (Game Object)

Default-Line (Material)

Shader: Legacy Shaders/Particles/Alpha Blended Premultiply

Figure 32. the options of the XR Ray Interactor component (left) and the XR interactor Line Visual as a separate component (right)

The mechanism and the visual are *decoupled*, as shown in Figure 32, as two separate components (the XR Ray Interactor and the XR Interactor Line Visual). This decoupling has two implications for our analysis. First, we cannot precisely know a game object's functional range, size, or scope simply from its appearance. Similarly, when grabbing an object with direct interaction, the object's 3D mesh size, in many cases, is not the size of its collider that functions in the collision detection. Thus, the only way to know the implementation and the trigger conditions under the hood is to experience and try the interactions many times. Second, from the design perspective, the visual form can employ additional information and freedom to ease the interaction. On this point, I have discussed examples such as the gun interface optimization (Krompiec and Park 2019) and the cases of eating and drinking consumables in *Population: ONE*. The ray interactor and the line visual by default provides two color states of the line to indicate whether the interaction with the line is valid or not.

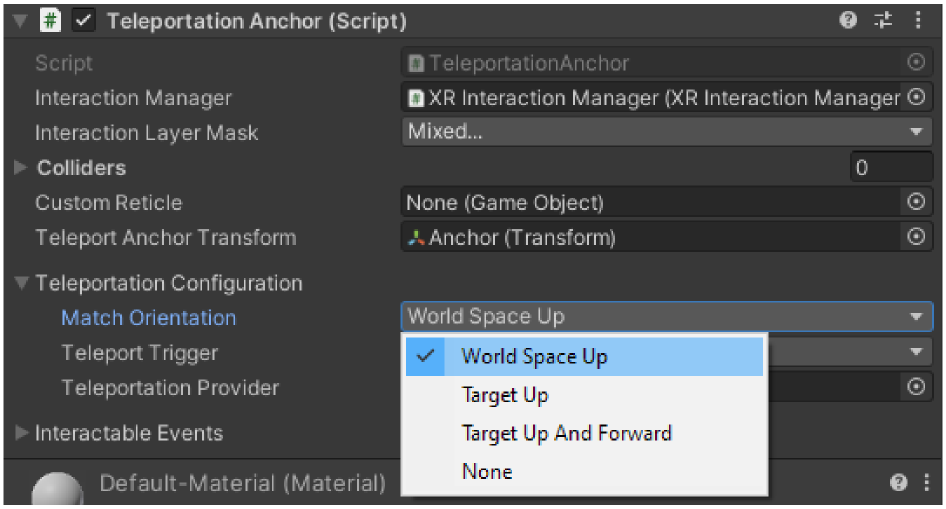
The *Line Type* option on the ray interactor component provides straight and curved line options for different use cases. The curved ray is commonly used for teleportation. I favor the projective curve for teleportation. It *simulates* a fishing pole effect and the jumping metaphor and *efficiently* points to a ground destination by rotating the controller up or down. On the other hand, the straight line can be more efficient for pointing and selecting targets with surfaces vertically laid out in front of the users, such as large objects and UI menu screens.

The *Anchor Control* option on the ray interactor allows the user to move the attached object of the ray interactor. For example, upon selecting a cube from a distance, the user can pull the cube close to the hand and rotate it horizontally. However, manipulating selected objects requires joystick controls that *conflicts* with the move and turn controls. We will come back to this problem shortly in the next subsection of control schemes.

Teleportation is another interaction that builds on the ray interactor. XRIT implements a locomotion system that controls the XR rig and provides teleportation, continuous movement and turn, and snap turn. Teleportation has a few options that reveal the design values of simulation and efficiency.

The first option is to specify an *anchor* within the target area to force the landing position. This is an efficient approach to navigate players to specific jump points. Steed et al. argue that it underscores the important *trade-off* between precision and fun, as free teleportation requires extra skills to move precisely to the location that matters to perform further tasks and can frustrate players if they have to reposition themselves repeatedly (Steed et al., 2021). In the introduction of the dissertation, I also mentioned a disadvantage of teleportation is in micro-adjustment.

Besides matching the position, XRIT also allows matching orientation to the target orientation after teleportation. Steed et al. find that in the seated mode, where players may not be able to turn all the way around, games such as *The Room VR* subtlely predetermines the orientation in addition to the position after teleporting to anchors. I agree that matching the orientation is also an efficient design, given the context of the game is about testing players' puzzle-solving (which can already be frustrating) abilities in structured, artefact-rich room environments rather than mechanically position themselves.



Teleportation Anchor (Script)
Script: TeleportationAnchor
Interaction Manager: XR Interaction Manager (XR Interaction Manager
Interaction Layer Mask: Mixed...
Colliders: 0
Custom Reticle: None (Game Object)
Teleport Anchor Transform: Anchor (Transform)
Teleportation Configuration
Match Orientation: World Space Up
Teleport Trigger
Teleportation Provider
Interactable Events
Default-Material (Material)

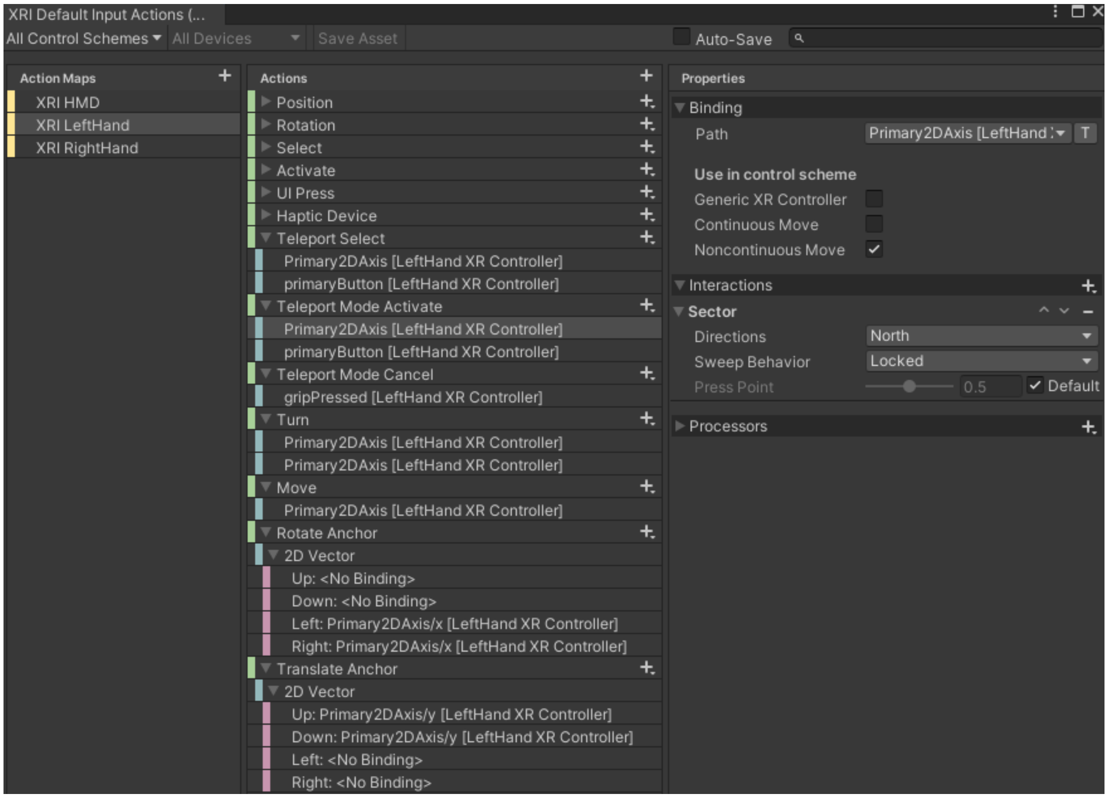
Figure 33. The Teleportation Anchor component and options

The Match Orientation option covers the target's forward direction but also the up direction for situations such as teleporting to a tilted plane or a hillside. For example, when teleporting from a flat ground to a tilted plane, the player will face forward parallelly with the plane.

2.3.3.3. Input actions and control scheme

XRIT comes with a default control scheme for the HMD and hand controllers. Figure 34 shows the interface to configure the input actions and control schemes.

This interface expands from left to right. Each Action Map contains a set of actions, and each action contains the input bindings that control the action. This system reflects what I meant by the *semantics* and the *techniques* of the action. The actions listed here are described by their semantics, and the input bindings are the techniques to perform the action. For example, the left hand has several teleport actions, turn, move, rotate and translate anchors (grabbed objects). When the user pushes the joystick in the *North* direction, the system triggers the *Teleport Mode Activate* action. Next, the user points the curve of the ray interactor coming from the left controller to select a teleportation target. Finally, when the user releases the joystick, the system triggers the *Teleport Select* action to teleport the user to the selected target.



Action Maps
- XRI HMD
- XRI LeftHand
- XRI RightHand

Actions
- Position
- Rotation
- Select
- Activate
- UI Press
- Haptic Device
- Teleport Select
  - Primary2DAxis [LeftHand XR Controller]
  - primaryButton [LeftHand XR Controller]
- Teleport Mode Activate
  - Primary2DAxis [LeftHand XR Controller]
  - primaryButton [LeftHand XR Controller]
- Teleport Mode Cancel
  - gripPressed [LeftHand XR Controller]
- Turn
  - Primary2DAxis [LeftHand XR Controller]
  - Primary2DAxis [LeftHand XR Controller]
- Move
  - Primary2DAxis [LeftHand XR Controller]
- Rotate Anchor
  - 2D Vector
    - Up: <No Binding>
    - Down: <No Binding>
    - Left: Primary2DAxis/x [LeftHand XR Controller]
    - Right: Primary2DAxis/x [LeftHand XR Controller]
- Translate Anchor
  - 2D Vector
    - Up: Primary2DAxis/y [LeftHand XR Controller]
    - Down: Primary2DAxis/y [LeftHand XR Controller]
    - Left: <No Binding>
    - Right: <No Binding>

Properties

Binding
Path: Primary2DAxis [LeftHand :

Use in control scheme
Generic XR Controller ☐
Continuous Move ☐
Noncontinuous Move ☑

Interactions
Sector
Directions: North
Sweep Behavior: Locked
Press Point: 0.5 ☑ Default

Processors

Figure 34. XR Interaction toolkit's default control schemes and input actions

As I mentioned earlier, the Move and Turn actions use the same joystick inputs as the Rotate and Translate anchor actions do. To resolve this conflict, developers can create control modes as state machines to enable and disable different input actions. In the example project[11] of XRIT, the system switches between *select*, *teleport*, and *interact* states to manage what sets of input actions are enabled on each hand controller in each of the states to avoid their conflicts.

This example reveals a common design challenge in VR games. When the number of game interactions increases to a certain threshold, the inputs of the two controllers fall short to accommodate them simultaneously. Also, the control scheme can be a place to amplify and resolve the tension between simulation and efficiency, as some actions may have to be implemented with motion or button inputs due to the limited inputs.

---

[11] https://github.com/Unity-Technologies/XR-Interaction-Toolkit-Examples/tree/master/VR

In the last case, we will take a close look at the platform that this study will use – the *Oculus Rift S* VR platform, to review its hardware configurations and broader contexts surrounding its publishment.

2.3.4. This Case: Oculus Rift S VR Platform

It is rare enough in the human history of computing that fundamentally new interaction platforms are commercialized, which can change how massive users communicate and entertain in virtual worlds and create huge markets. Gaming is one of the biggest applications of VR. For example, the worldwide shipments of VR gaming headsets reached over five million in 2018 (Statista, 2020), contributing to a US$7.9 billion global VR market of the same year (Research and Markets, 2019). In 2020, the global VR market was valued at US$15.81 billion, and it is expected to grow at an annual rate of 18% from 2021 to 2028 (Grand View Research, 2021).

Among all VR headsets, Facebook's Oculus VR headsets (Oculus Quest, Quest 2, Rift, and Rift S, etc.) have become one of the most popular brands, and nearly half of the VR gamers on the Steam online game store use Oculus (Statista, 2020). At the Oculus' Connect conference in 2017, Facebook CEO Mark Zuckerberg set an ambitious goal of getting a billion people in VR. Among the Oculus VR product line, the Rift is the first generation headset, and the Rift S is an enhanced version with better graphics and display. The company's trend is moving to wireless and untethered devices that do not need to connect to a powerful PC, starting with the Quest headset and following up with its second generation. Still, many current VR games with high-quality graphics need to run on a VR-supported PC, and that is why I choose the Rift S as the target platform that can play most PC VR games. Regardless of the headset, the controllers and their inputs have remained overall unchanged.

Figure 35. Oculus VR headset and motion controllers with annotations on the three major physical inputs

The Oculus VR controller, named Oculus Touch controller, features an ergonomic design for hands with three mostly used physical inputs, as shown in Figure 35. It has a joystick that can be rotated in any direction and pressed as a button for the thumb finger, a trigger button for the index finger, and a grip button for the middle finger. In addition, it has two action buttons and a system button. On the left hand, the two action buttons are labelled as X button and Y button, while on the right hand, they are labelled as A button and B button.

On its developer webpage, Oculus provides a design guideline for mapping the controller inputs to their functions in the game:

> **“Button Functions in Experience**
>
> A/B/X/Y: Title-specific actions
> Oculus button: Reserved, not available to developer
> Menu button (≡): Bring up or dismiss a pause or in-app menu
> L/R Grip Button: Picking up/grabbing/releasing objects
> L/R Trigger: Firing weapons, selecting objects, etc.
> L/R Thumbstick: Locomotion, teleport, snap turns, etc.
>
> Notes:
> 1. Using the Menu button (≡) for menus is strongly recommended, as it provides a more consistent experience across titles.

> 2. In titles where the user can throw held objects, we recommend using the Trigger instead of the Grip Button to perform those actions.
>
> 3. For games using first person locomotion and snap turns, we recommend using one Thumbstick (such as left) for movement, and the other for snap turns." (Oculus VR, 2017)

Although this is a useful reference for developers, it is only a general guideline. In the comments on this webpage, developers further discussed diverse options on mapping the object interactions of picking up and throwing. This guideline also does not include motion and gesture-based interactions. Therefore, there is still more work to be done to understand the more complex design space of control schemes and their impact on game interactions.

If VR wants to transition into a large social and gaming platform that can connect one billion users like what Oculus claimed, besides advancing the hardware, its interaction design must be easy-to-use, like what iPhone has done on gesture-based screen interactions for mobile phones. Historically, videos games have motivated advancements in computing hardware and interaction design. Therefore, by examining the emerging VR game interaction approaches, we can inform future iterations of VR. Although currently, the major brands of VR hardware all have the same layouts and number of inputs on the controllers, it is still hard to say what VR controllers will look like in the future.

Egliston and Carter suggest that Facebook envisages VR to be integrated into its wider social media platform as an everyday repertoire of communication, and this Oculus imaginary departs from existing narratives of VR as an immersive gaming media (Egliston & Carter, 2020). This may be a reason why the design guideline only covers minimal basic locomotion and object interactions. However, if the Oculus VR platform is more invented as part of social media, how sufficiently will it support gaming? We will see this through examples of game interactions in later analysis.

2.4. Literature Review Summary

In this literature review, I connect theories from different areas of study and identify simulation and efficiency as two "meta" design values that can guide the design of game interactions in VR. I use the following list to summarize how I find game scholars have addressed and emphasized various perspectives of simulation and efficiency.

Games as simulations

- A simulation is accurate and detailed, while a game can deliberately suppress details to be clear and stylized to convey the designer's message (Crawford, 1984).
- Games are often stylized simulations (Juul, 2005).
- "A simulation is a representation of a source system via a less complex system" that reflects subjective design choices (Bogost, 2006).

Levels, measurements, and design strategies of simulations for game interfaces and interaction

- Beyond iconic (realistic) simulations are indexical (e.g. the Diablo's inventory slots) and symbolic (e.g. the dice) simulations (Dormans, 2011).
- The "literal-metaphorical-abstract" scale of interface functional narrativization (Tanenbaum & Bizzocchi, 2009)
- Behavioral mimicking and behavioral metaphor, where the physical behaviors of the player's gameplay interactions mimic, or suggest a metaphorical connection to, real-life actions. (Bizzocchi et al., 2011)
- High interaction fidelity (IF), "the objective degree of exactness with which real-world interactions can be reproduced" (McMahan, 2011).
- Use substitutions and approximations for physical challenges to suspend disbelief for moderate IF (Rogers et al., 2019).

Efficiency as an aspect of usability, playability, and ludic interaction

- Efficiency is a function of the accuracy and completeness in achieving certain goals and the resources expended, and indicators include task completion time and learning time (International Organization for Standardization, 1998).

- Efficiency relates to the functional component of playability - how well the input and output mechanism can achieve gameplay requirements (Järvinen et al., 2002), which also determines the learning curve (Rouse III, 2001).
- Efficiency is reflected by criteria of the game flow framework, including concentration, player skills, control, clear goals, and feedback (Sweetser & Wyeth, 2005). See section 2.1 for detailed descriptions of these criteria.
- Efficiency, adapted from the definition of ludic efficiency (Tanenbaum & Bizzocchi, 2009), refers to the extent to which the interaction design eases or hinders the player's attempt to perform the game action.
- Efficiency can explain why sometimes *low* interaction fidelity can be better in the experience.

Among the perspectives on simulation, I find the "literal-metaphorical-abstract" scale, behavioral mimicking and behavioral metaphor, and the framework for interaction fidelity analysis provide practical measures to evaluate the value of simulation within the game interaction design, while other perspectives are more conceptual. Furthermore, I find gaps and limitations within these design-related frameworks on simulation when applying them to VR game interactions and opportunities to extend them. Noticing that simulations seem to be over-emphasized but is not the only thing that matters and that efficiency often appears in the discussions of simulation, I consider whether simulation can conflict with efficiency, another fundamental design value that is often taken for granted but can open up many design details in practice.

With the measures of simulation and efficiency from the literature, I use the two-dimensional framework (Figure 36) to explore how the two values co-function in VR interaction examples.

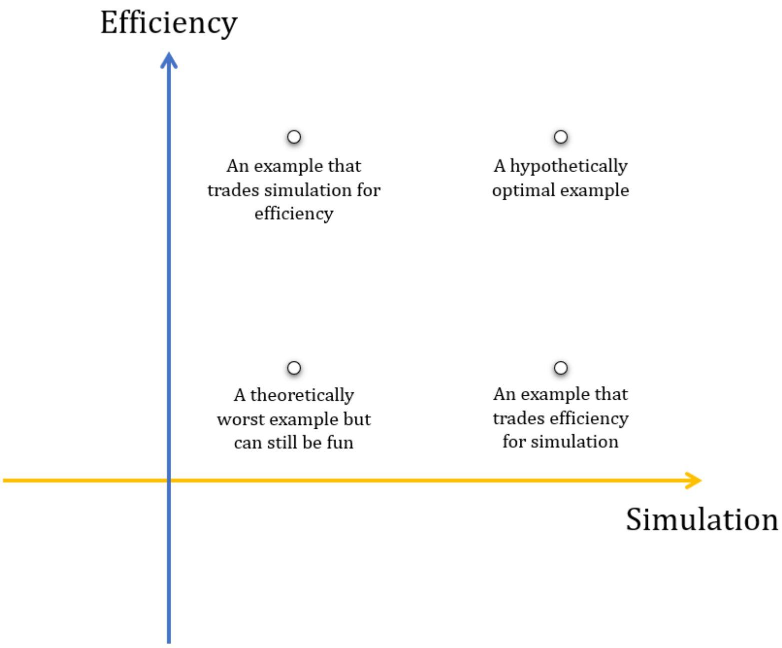


Figure 36. The Simulation - Efficiency framework with four representative examples

Platform and game context

Although this framework can position these examples and represent their value tensions, what is more meaningful is the reasons why they are in these positions and the design knowledge of resolving their tensions. Therefore, I further investigate two places connected to VR game interactions – game context and platform. Using the phrase "VR game interaction", I literally mean that the interaction is based on the VR platform and within the game context.

As shown in the conceptual map in Figure 37. The interaction level connects the underlying platform aspects and the game context that the interaction serves.Figure 37, the interaction is an intermediate level between the platform that supports it and the game context in which it takes place. The interaction involves the game action as its core component, and signifiers and feedback are designed to facilitate the interaction. For the game action, I further use semantics and techniques to describe the meaning and the implementation. Research into VR interactions has primarily studied locomotion techniques for basic actions such as move and turn. However, many meaningful game interactions are still underexplored, such as combat and object interactions, let alone their design details involving the platform and game context.

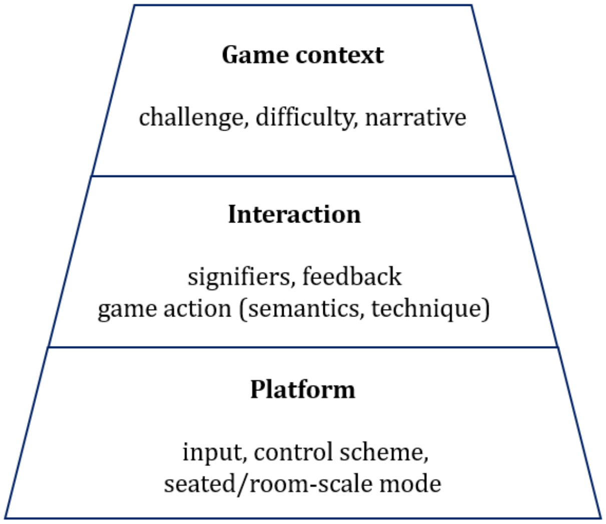


Figure 37. The interaction level connects the underlying platform aspects and the game context that the interaction serves.

For the platform aspects, I use platform studies as a lens and reflect on my previous work on XR platforms, highlighting factors such as trigger conditions, control schemes, and the physical configuration of the player and the play space that afford and constrain the interaction and experience. In my discussions, I also show that the game context can affect the design values.

Game context adds to the complexity and the number of factors in this design space. It involves the challenge, the difficulty, and the narrative, to name a few, as identified in Caroux et al. (2015)'s framework of player-game interaction. Many studies on VR interaction techniques use prototypes in lab settings, some of them are based on a small number of design cases, but most of them do not focus much on the game context. Although it may sound like that game context is an infinite space varying from game to game, it can still be comprehensible through patterns and themes, such as time urgency and the number of interactions needed to perform. In this study, I will use close reading to analyze examples of interaction within the context of the gameplay to discuss how the context affects the balance of the two design values.

In sum, simulation, efficiency, game interaction, platform, and game contexts are the core concepts and keywords of this literature review. The platform aspects and game context are two additional sources besides the game interaction itself to locate and explain the potential value tensions between simulation and efficiency in the design of VR game interactions. Through a close reading of modern VR games, we can identify emerging approaches that achieve the optimal design and extend our understanding of existing theories on game interaction – particularly frameworks on narrative and embodied interface and interaction fidelity.

3. Data and Methods

This chapter will be more straightforward to explain the data collection process and the methods I will use to analyze the data. Section 3.1 will present my dataset and the data organization using the semantics of actions as the categories. Section 3.2 will discuss case studies, close reading, and the analytical lenses.

3.1. Data-Driven Approach and VR Interaction Dataset

This section will first give a brief overview of the recent case studies and data-driven approaches emerging in VR interaction. I will discuss the interactions and perspectives in these studies and the benefits of having a dataset of interaction examples.

3.1.1. Recent Case Studies and Data-driven Approaches

While early VR research mainly studied prototypes in lab settings, case studies and analyzing datasets of modern game interaction examples have become popular methods in recent years.

For example, the Locomotion Vault dataset[12] presents over a hundred examples and visualizations of locomotion techniques in VR and 3D games. Their analysis highlights "an inherent trade-off between simulation sickness and accessibility" across different locomotion techniques (Di Luca et al., 2021). Conceptually this is similar to my study, but I

---

[12] Link to the dataset: https://locomotionvault.github.io/

will cover a wider variety of interactions and focus on a different trade-off framed in terms of design values.

Steed et al. identified several 3DUI design patterns and novel implementations, including interaction techniques for selection, manipulation, locomotion, system control, and adjustments (Steed et al., 2021). Some of their analytical lenses, such as accessibility, diegetic interfaces, and physical interactions, are similar to my concerns of simulation and efficiency. What I cover more in this study, beyond the sorts of system control, menu UIs, and general VR interactions, is the interactions that are more related to the gameplay activities and those that define the theme and genre of the games, such as combat interactions in FPS games and RPGs.

I am cautious about calling my findings “design patterns” because design patterns in game design (Bjork & Holopainen, 2004; Kreimeier, 2002) tend to have a shorter lifespan than those in technical and industrial design fields. Design patterns in architecture (Alexander, 1977) and software design (Gamma, 1995) often apply to highly structured and well-defined problems that need to be solved, which may not be the case in game design. Furthermore, design patterns are a high bar to meet, with the goals of reusable solutions and golden rules that often need years of practice to prove. However, VR and VR games are still rapidly developing and only have a short history. Therefore, I prefer using “emerging approaches” to address the best practices of VR game interactions in my study.

Besides the most commonly performed interactions, such as locomotion interactions, some recent studies have focused on a specific interface or function in VR. For example, Danyluk et al. analyzed twenty-five examples of “worlds in miniature” interfaces in XR and identified several design dimensions that could influence their uses (Danyluk et al., 2021). Tanenbaum et al. examined expressive non-verbal communication from an inventory of ten social VR platforms and identified several design strategies for movement, facial control, and gesture (Tanenbaum et al., 2020). Overall, they found that the controls for facial, posture, gesture and other subtle emotional expressions either have a low interaction fidelity or are nonexistent. They often need several levels of navigation in the game menu and are not direct interactions. Sometimes they are even automated. They cannot be performed simultaneously with other interactions.

The scarcity of these subtle emotional and gestural controls reveals the limitation of the current commercial VR platform. Because only the player's head and two hands are tracked, it is not easy to directly map those non-verbal bodily interactions to the VR controllers. If more parts of the player's body are tracked, it will be more technically feasible. In current social VR platforms and multiplayer games, we often see unnatural arm angles and rigid bodies of player characters.

From a methodological perspective, the benefit of these case studies and the data-driven approach is straightforward. They are an excellent way to understand the design space, identify the gaps, and extract design knowledge from the emerging interaction examples. As Di Luca et al. (2021) suggest, academic research has been slow to follow up with the proliferation of interaction techniques presented by the VR games in the market. The data-driven approach is believed to address the existing challenges and support the creation of new interaction methods to fill the gaps in current VR interaction.

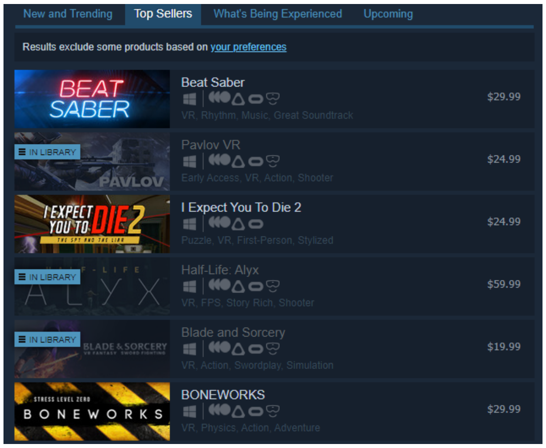
3.1.2. Data Collection and Game Selection Criteria

My data collection is a multi-step and iterative process, but it generally follows the three steps described below.

Step 1: Scope – focus on popular FPS and RPG titles

First, I choose to focus on two broad themes or games *genres* – FPS and RPG. They feature a wide variety of game interactions that are mapped to bodily actions. I exclude games with only several simple actions unless they provide novel and representative examples. Game genres such as rhythm games and sports games focus less on the character and combat and are playful less because of the richness of interactions than their unique game mechanics. Therefore, most non-FPS or RPG titles are not of this study's particular interest.

Among FPS and RPG games, I further scope down to those with high *popularity*, which I primarily rely on the "Top Sellers" list on the Steam store after applying the "VR" tag in the search (Figure 38). I also identify popular game titles from Youtube gameplay videos uploaded by VR enthusiasts and reviewers in the community (Figure 39).



Results exclude some products based on your preferences

Beat Saber	VR, Rhythm, Music, Great Soundtrack	$29.99
Pavlov VR	Early Access, VR, Action, Shooter	$24.99
I Expect You To Die 2	Puzzle, VR, First-Person, Stylized	$24.99
Half-Life: Alyx	VR, FPS, Story Rich, Shooter	$59.99
Blade and Sorcery	VR, Action, Swordplay, Simulation	$19.99
BONEWORKS	VR, Physics, Action, Adventure	$29.99

Figure 38. Part of the "Top Seller" list of VR games on the Steam store

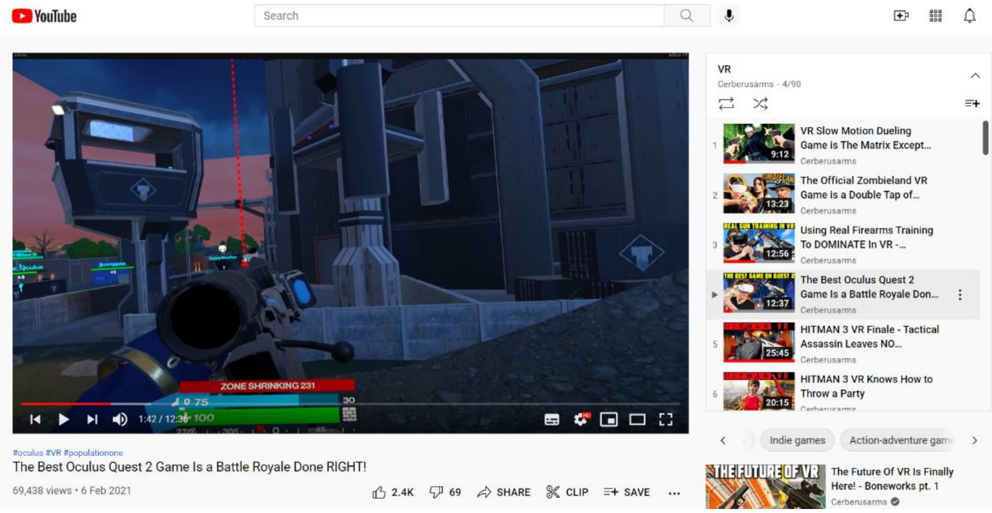
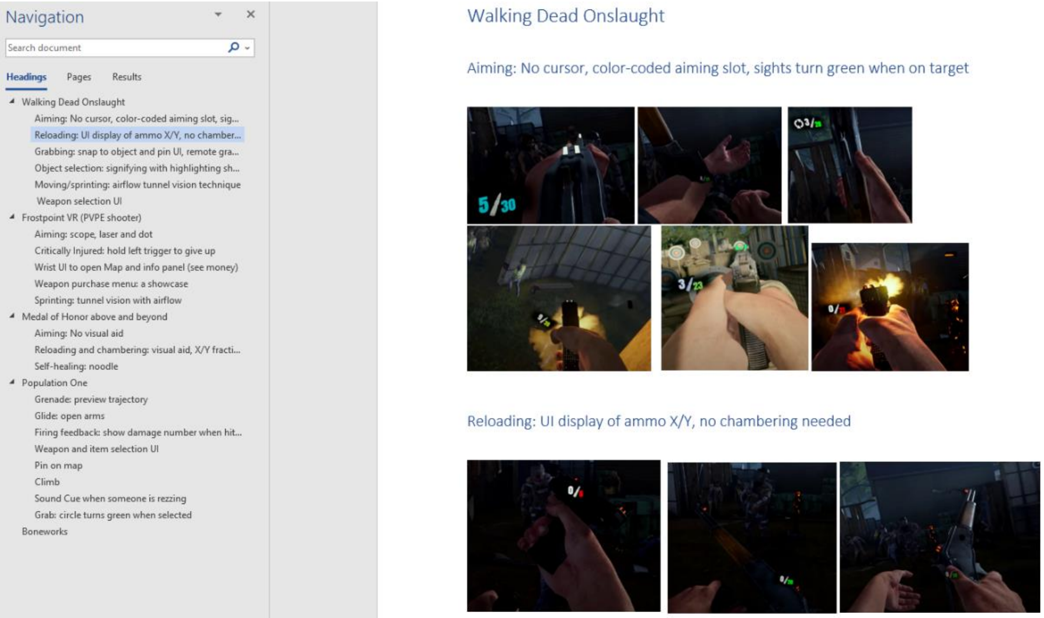


Figure 39. One of the YouTube channels with gameplay videos of popular VR games

Step 2: Raw data - notes on the game interaction design

Next, I start playing some of the games and watching videos of others playing them. Meanwhile, I take notes on the game interactions by taking screenshots and writing down a general category of what an interaction belongs to and what design features they have that reflect the values of simulation or efficiency. The notes are laid out in a Microsoft Word document with text descriptions of the game, action, design details, and screenshots, and they become my raw data, as shown in Figure 40. The screenshots are taken from gameplay videos and organized in file folders (Figure 41).



- Walking Dead Onslaught
  - Aiming: No cursor, color-coded aiming slot, sig...
  - Reloading: UI display of ammo X/Y, no chamber...
  - Grabbing: snap to object and pin UI, remote gra...
  - Object selection: signifying with highlighting sh...
  - Moving/sprinting: airflow tunnel vision technique
  - Weapon selection UI
- Frostpoint VR (PVPE shooter)
  - Aiming: scope, laser and dot
  - Critically Injured: hold left trigger to give up
  - Wrist UI to open Map and info panel (see money)
  - Weapon purchase menu: a showcase
  - Sprinting: tunnel vision with airflow
- Medal of Honor above and beyond
  - Aiming: No visual aid
  - Reloading and chambering: visual aid, X/Y fracti...
  - Self-healing: noodle
- Population One
  - Grenade: preview trajectory
  - Glide: open arms
  - Firing feedback: show damage number when hit...
  - Weapon and item selection UI
  - Pin on map
  - Climb
  - Sound Cue when someone is rezzing
  - Grab: circle turns green when selected
- Boneworks

Walking Dead Onslaught

Aiming: No cursor, color-coded aiming slot, sights turn green when on target

Reloading: UI display of ammo X/Y, no chambering needed

Figure 40. Raw data and notes for analysis laid out in a Word document

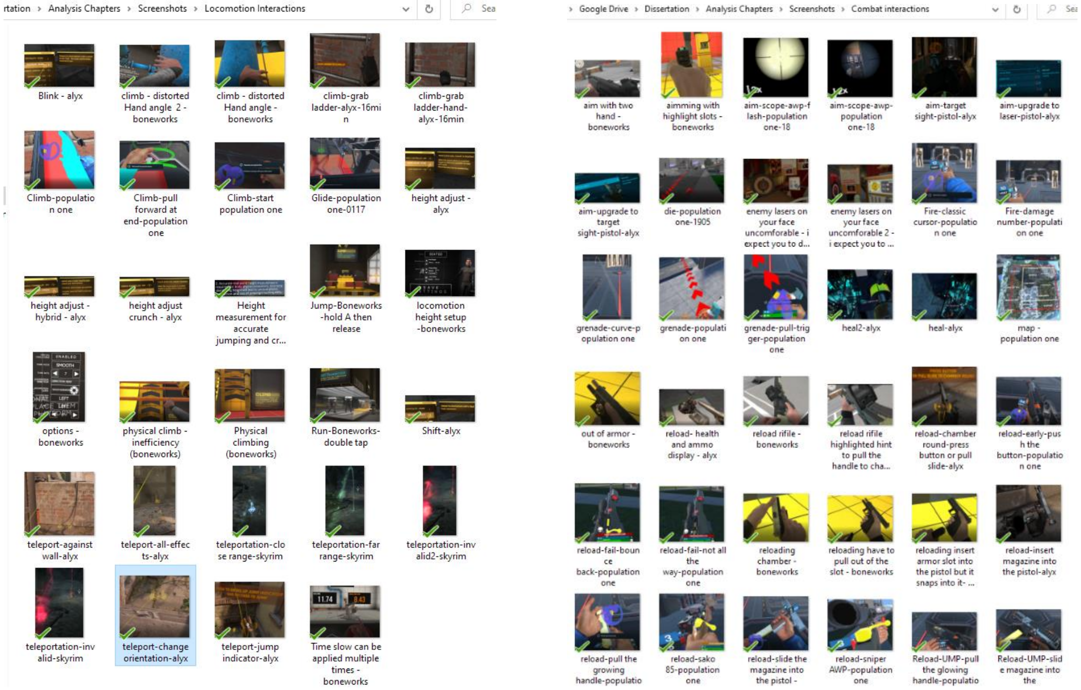


Figure 41. Screenshots of locomotion and combat interactions saved in folders

Step 3: Fine process – index the examples with game actions, add details and control schemes

While the raw data use the game title as the first-order index to organize the data, in this step, I transpose the data to use game actions, such as aiming and reloading, as the index and group the examples from different games under the same game action. I remove examples that show duplicated designs. For each example that demonstrates a unique design, I will play the game to experience the interaction myself and add more details to its description. I will also record its control mapping, which usually cannot be seen from others' gameplay videos.

Step 4: Analyze, iterate, and stop with saturation

Once I have multiple examples of the same game action, I write down the analysis that focuses on my research questions' ideas. I discuss how each example reveals the value tensions between simulation and efficiency and the aspects of the game content and the platform that amplify or reconcile them.

I conduct the analysis and iteration of data collection simultaneously. I keep searching for examples to explore the design space until I find saturation among the examples. While analyzing an example, I sometimes would collect other examples as I think of similar or alternative designs within them. As the data collection and analysis continues, I gradually form a dictionary of game actions and their design examples and variations. Finally, I organize them under broader categories presented in the following chapters.

3.1.3. Dataset, Categories of Actions, and List of Games Reviewed

The data are primarily screenshots stored in folders on my desktop computer. These screenshots are taken from a collection of gameplay video recordings, which I store in an external hard drive because they are over 200GB in size. Several screenshots are taken from online videos, which I provide the link to the respective video in the image's caption when I use them as examples in the analysis chapters. In addition, I keep a spreadsheet of how the controller inputs are mapped to various game actions in different games for ease of reference in the discussion. Each screen is carefully taken to select an area that captures the essential elements in the interaction design.

The dataset is finally organized by the categories of actions, as shown in Figure 42.

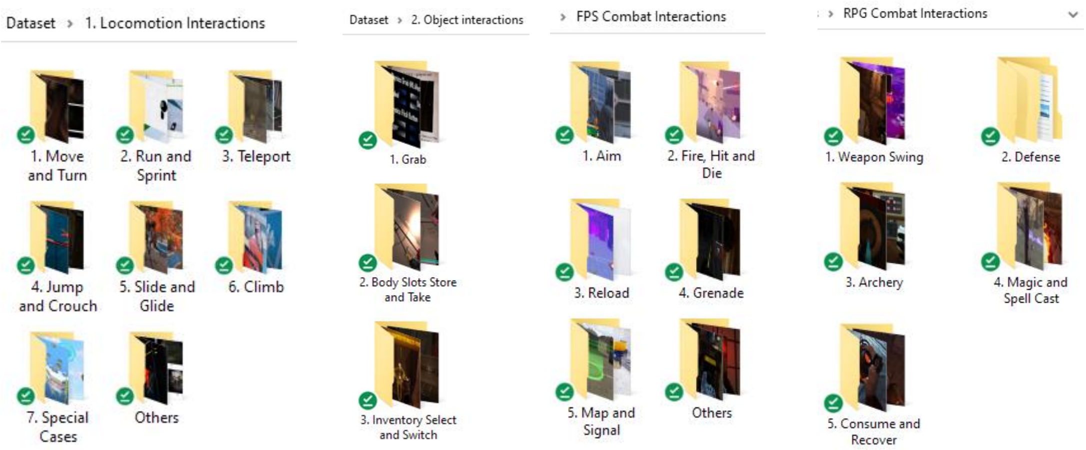


Figure 42. Screenshot data are finally organized in folders by categories of actions

I call this dataset the VR Game Interaction dataset. The four high-level interaction systems are 1) Locomotion Interactions, 2) Object Interactions, 3) FPS Combat Interactions, and 4) RPG Combat Interactions. Each of these four categories further contains examples grouped by the semantics of action.

The Locomotion Interactions include 1) Move and Turn, 2) Run and Sprint, 3) Teleport, 4) Jump and Crouch, 5) Slide and Glide, 6) Climb, and 7) Special Cases.

The Object Interactions include 1) Grab, 2) Body Slots: Store and Take, and 3) Inventory Slot: Select and Switch.

The FPS Combat Interactions include 1) Aim, 2) Fire, Hit and Die, 3) Reload, 4) Grenade, 5) Map and Signal.

The RPG Combat Interactions include 1) Weapon Swing, 2) Defense, 3) Archery, 4) Magic and Spell Cast, and 5) Consume and Recover.

These semantic categories of actions cover a wide range of commonly performed interactions in FPS and RPG games. They break the complicated action systems into modules for ease of analysis. In chapters 4, 5, 6, and 7, I will use the simulation and efficiency framework to analyze each of the above modules of interaction examples.

These interaction examples are from the following list of games (Table 1). The list contains 26 VR games of the FPS and RPG genres.

Table 1. List of reviewed VR games in this study

Game	**Developer**	**Publisher**	**Year**
Blade and Sorcery	WarpFrog	WarpFrog	2018
Boneworks	Stress Level Zero	Stress Level Zero	2019
Budget Cuts 2: Mission Insolvency	Neat Corporation, Fast Travel Games	Neat Corporation	2019
Echo VR	Ready At Dawn	Oculus Studios	2017
Frostpoint VR: Proving Grounds	inXile entertainment	Thirdverse	2020
GORN	Free Lives, 24 Bit Games Pty. Limited	Devolver Digital	2017
Half-Life: Alyx	Valve Corporation	Valve Corporation	2020
Hellsplit: Arena	Deep Type Games	Deep Type Games	2019
Hot Dogs, Horseshoes & Hand Grenades	RUST LTD.	RUST LTD.	2016
I Expect You To Die 2	Schell Games	Schell Games	2021
Ironlights	E McNeill	E McNeill	2020
Legendary Tales	Urban Wolf Games Inc.	Urban Wolf Games Inc.	2021
Onward	Downpour Interactive	Downpour Interactive, Coatsink	2016
Pavlov VR	Vankrupt Games	Vankrupt Games	2017
Population: ONE	BigBox VR, Inc.	BigBox VR, Inc.	2020
Sairento VR	Mixed Realms, Swag Soft Holdings Pte Ltd	Mixed Realms, Perpetual Europe	2016
Shadow Legend VR	VitruviusVR	VitruviusVR	2019
STRIDE	Joy Way	Joy Way	2020
SwarmVR	Greensky Games, Inc.	Greensky Games, Inc.	2021
SWORDS of GARGANTUA	Thirdverse	Thirdverse	2019
Tales of Glory 2 – Retaliation	BlackTale Games	BlackTale Games	2020
The Climb 2	Crytek	Crytek	2021
The Elder Scrolls V: Skyrim VR	Bethesda Game Studios	Bethesda Softworks	2017
The Walk Dead Onslaught	Survios	Survios	2020
Until You Fall	Schell Games	Schell Games	2020
Waltz of the Wizard: Natural Magic	Aldin Dynamics	Aldin Dynamics	2021

3.2. Case Studies, Close Reading and "Hacking" Game Interaction Design

In this section, I will explain more on the methods and methodology that I will use to interpret the data and address the research questions. I will also discuss the specific analytical perspectives I will take in the analysis. I do so to explain why the methods are rigorous and to detail my thought process, the lenses, and the validity of this inquiry.

Because of the nature of my data and the practice of interpreting them, this investigation into game interaction design uses *case studies* on the high level and *close reading* on the level of each case. On the high level, this study is akin to the case study of Flanagan and Nissenbaum's work on values at play in digital games, in which they review cases of game design and group them to discuss each specific sociocultural value (Flanagan & Nissenbaum, 2014). This study is also like the abovementioned, recent VR interaction studies taking a data-driven approach. The practice of theorizing a framework or discussing patterns and themes from multiple cases is the same, and it is commonly called "case studies". In each case, however, the practice of the researcher interpreting the data to reflect on the interaction design is a practice using the close reading technique.

Close reading is a hermeneutic method that game scholars have adapted from its use in literary theory. Readers can refer to some foundational work for detailed theoretical concerns around hermeneutic inquiry and close reading and their adaption to game studies (Bizzocchi & Tanenbaum, 2011; Tanenbaum, 2015a, 2015b). Here, I focus more on stating my analytical lenses and addressing potential challenges of close reading discussed in those foundational works and how to deal with them in this study.

3.2.1. Analytical Lenses

As I have discussed throughout the writing so far, the primary goal of this dissertation is to explore the design space of VR game interactions by examining the following key dimensions: *simulation*, *efficiency*, *game actions*, *context*, and *platform*, and in particular, the potential value tensions between simulation and efficiency. I use the following prompts and cheat sheets to guide my analysis.

Analytical lens #1: Simulation

This lens is grounded in the frameworks of narrative and embodied interaction and interaction fidelity, which I use as the primary measurements of simulation.

1. How does the game interface's visual, audio, and haptic *forms*, *signifiers*, and *feedback* simulate those of its physical counterpart?
2. How does the game interface's *function* simulate its physical counterpart's function?

3. To what extent does the game action mimic the physical *behavior*? To what extent are *body segments*, *motion*, *forces*, and *control functions* involved in the game action symmetric to the physical behavior?

Analytical lens #2: Efficiency

This lens is grounded in ludic efficiency, definitions of efficiency as an aspect of usability and playability, and the game flow framework.

1. How does the game interface's visual, audio, haptic forms, signifiers, and feedback ease or hinder my attempt to perform the game action?
2. How simple or problematic is the physical behavior and control?
3. How much time[13] does it take to perform the game action?
4. How strict is the triggering condition of completing the game action?

Analytical lens #3: Value tensions, context and platform

This lens extends the above ones and is grounded in frameworks of the duality of simulation and efficiency, value tensions, player-game interaction and platform studies.

1. Where does the example fall onto the two-dimensional framework of simulation and efficiency? Does the interaction design achieve only simulation or efficiency, but not both simultaneously? Or, does the interaction design achieve both values? Either way, try to explain why it is the case and identify the factors that may affect the design choice.
2. How well does the game interaction fit into the game's overall theme and gameplay context? Do any aspects of game context rationalize the design of prioritizing either simulation or efficiency?
3. How does the control scheme reveal the value tensions? Do the controller inputs become a limited resource for accommodating all the game actions in this game? If so, among the actions that can be assigned with different controls, which actions achieve simulation and which ones are efficient?
4. Do the seated and the room-scale play modes affect the choice between the design values?
5. What are other constraints and limitations of the platform that can explain the value tensions?

---

[13] I won't be counting how many seconds to record a numerical value of the time, but only to compare the time of performing two actions relatively based on the steps of operating their controls and report how I feel about their time and complexity.

Many of these prompts focus on the formal elements of the interaction. Game scholars Lankoski and Björk propose a method called formal[14] analysis (Lankoski & Björk, 2015). Like lines and colors in visual art and rhythm in poetry, games contain formal features such as game elements, rules, and goals. To conduct formal analysis, they identify a set of formal primitives – components, actions, and goals – and explore their relationships in the game system. In practice, this method is based on close reading the game. The kinds of close reading that I mentioned from the foundational works involve a higher degree of *subjectivity* in connecting the forms of game narratives to player experience and aesthetics in design, while Lankoski and Björk claim their method to be more focusing on the objective, player-independent elements.

I am particularly interested in leveraging close reading to "hack" or reverse-engineer the game interaction design to extract best practices and design knowledge from emerging approaches. Because video games are commercial products in the industry, many of the best practices and design knowledge are private and exclusive to the game development teams and often treated as the selling points in the competitive video games market. The reverse-engineering approach is usually how professional developers learn from each other by dissecting what competitors have done based on the forms of their work. This is like what hackers do to understand the underlying logic of a computer system or program based on its manifested functions and behaviors. In terms of game interaction, however, the barrier of knowing how things work behind their forms is relatively low. Although one may never know the source code, most of the logic and function is reflected by the form of the interaction itself. Therefore, I believe this reverse-engineering lens of close reading on game interactions is a productive approach to examine game interaction. Specifically, as I mentioned, formal elements including controls, trigger conditions, signifiers and feedback can be figured out using this reverse-engineering lens.

3.2.2. Challenges and Validity

---

[14] As noted by the authors, the word formal here refers to the forms of work, not formalism in mathematics or philosophy.

There are a few common challenges of close reading that game scholars often have to address to clarify the rigor. Bizzocchi and Tanenbaum identified three challenges of close reading unique to the digital media: *indeterminacy*, *scope*, and *difficulty* (Bizzocchi & Tanenbaum, 2011).

Indeterminacy: the game content can vary based on which player is playing the game. However, in most cases, game actions do not have a high degree of variance. They are designed to be independent of who is playing the game. I seek to browse all the designed forms of a game action in my data collection, although I only discuss the representative ones in the analysis. I treat game interactions as formal elements to avoid any potential issues of indeterminacy.

Scope: there may be cases where certain assets or variances of game actions are not unlocked until some progress has been made. Although some games can have hundreds of hours of content, most of their interactions are shown to players at a very early stage. In those rare cases where there are newly learned interactions in later game content, I will watch videos or speedily play the game to see the interaction.

Difficulty: because the measurement for efficiency is partially based on the time to complete the action, my skill and the difficulty of the gameplay may affect my judgement of how efficient a game interaction is. My skill to perform the interaction may also change during the gameplay after I get more used to it.

I take two strategies to minimize this potential challenge. First, I avoid using absolute time in seconds to ground my evaluation. Instead, I only compare time qualitatively and relatively based on the steps and complexity in performing the actions and operating the controls. For instance, the time of pressing a button is considerably much less than the time of performing an arm swing. Second, I try my best to practice it multiple times for any complicated bodily action so that I pass the learning stage and have a fair mastery of it. However, difficulty itself can be a measure of inefficiency. Some actions are inherently difficult to perform due to strict trigger conditions, small timing windows, and multiple steps. I will point out the objective challenges within the interaction design as part of the analysis.

Bias and Validity

Besides these specific challenges, the scholar's bias and the research's validity are two broader methodological concerns of close reading. Validity is a measure of how successfully the researcher has achieved the research goal, and its standards vary across different epistemological paradigms. For qualitative research, validation strategies include *clarifying the researcher's bias*, using *rich-thick descriptions*, and *triangulation* of multiple data sources (Creswell, 2003). Close reading is a kind of qualitative and constructivist inquiry and is primarily based on the researcher's own interpretation of the data. Therefore, clarifying the researcher's bias is essential to validate close reading (Tanenbaum, 2015a).

The bias in this study mainly lies in my judgement of to what degree a given game interaction has achieved the design values of simulation and efficiency, whether any tensions between the two design values exist, and how they are amplified or reconciled by elements of the game content and the platform. However, when I place examples onto the coordinate frame of simulation and efficiency, I do not intend to locate their coordinates precisely but rather to show an overall tendency and draw comparisons. I do not think their absolute positions are meaningful without the units that are difficult to define for each continuum. It is their relative positions and the explanation that is more meaningful. I acknowledge that if a different researcher comes in to do the analysis, they may have other rankings and reasoning that are not precisely what I produce in this dissertation, even if we use the same analytical lenses that I have set above.

However, the burden of proof in this dissertation is not describing any objective truth or rankings about VR game interaction that other people will all agree with. In fact, my bias itself is part of my subjective interpretation and what I hope to provide – an exploration and understanding of the design space of VR game interactions grounded in emerging examples. My goal is to show that we can better understand and design VR game interactions, which is messy and complicated, through a new conceptual framework that consists of the two meta design values - simulation and efficiency - and related aspects of game context and platform. This framework is developed both to be an extension of existing frameworks and from my observation of the VR game interaction dataset of the state of commercial VR games. In this sense, it is also a design reflection on what we have achieved in interaction design for VR gaming, which has a short history but will be a long-term development in the future. For each game action, I use multiple examples for triangulation, and for each example, I use rich-thick descriptions to explain the details of my evaluation. Using the dataset I have collected, this evaluation can be reproducible, and it can foster more discussion on this topic and guide future design practices.

3.3.3. Explaining the Charts: Subjectivity and Visuals

Although the above section has stated some of my potential bias of mapping the examples onto the framework, the visual forms on my charts can potentially trigger further questions and concerns about what they represent. Therefore, I added this section to explain more details.

Let me start by reemphasizing the subjectivity of the mappings on the simulation and efficiency axes. Why can they not be objectively positioned? Some of the criteria such as interaction fidelity and behavioral mimicking can be objectively measured, but my values of simulation and efficiency are combinations of these criteria, whose weight that I assign to the mappings are often affected by specific game contexts and cannot be objectively quantified. Also, because I want the framework to be an extension of the two existing frameworks, I follow their conventions of not problematically stating any objective mappings. The mappings onto the Ludic Efficiency and the Functional Narrativization scales are subjective and relative. I have also argued that the discrete levels of high, moderate, and low interaction fidelity are problematic in terms of vague thresholds and missing game contexts. Although the general audiences within the game community may tend to like objective statements and quantitative rankings, such as how people rate games on online stores, for academic rigor, I avoid potential drawbacks of objective and quantitative approaches and instead use subjective and qualitative analysis to focus on elaborating the insights and my own experience.

The subjectivity also gives me room to work around the visuals on my charts. Here I answer some potential questions on the charts that readers will see in the rest of the analysis. For contexts and discussion, I put my framework here again in Figure 43.

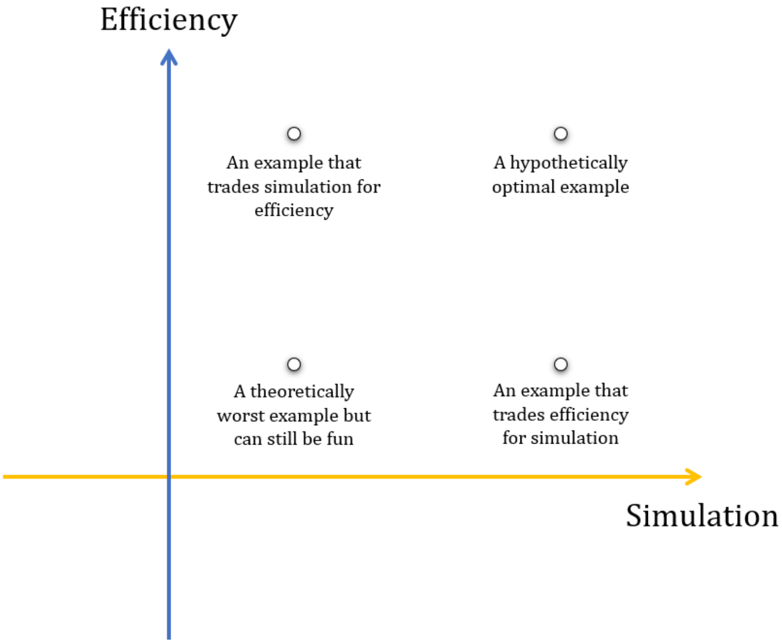


Figure 43. Mappings of examples on the simulation and efficiency framework

1. Why do the lengths of the simulation and efficiency axes vary in different game actions? It is mostly for laying out all the representative examples and having clear typography. Some actions, such as teleport and climb, have many examples that require me to extend the axes longer than what the above one has. Varying the length does not affect the *relative positioning* of the examples, which is what matters in my analysis. I do not find having a fixed length is particularly useful because comparing interaction examples across different semantic actions is not my focus.

2. Do the two axes have any units? No, the two axes also do not have units. There are only the zero point and the endpoint that serve as the theoretical extremes. The negative side of each axis is practically meaningless, at least in my analysis. I do not find any units that are can be meaningfully quantified for the values. The lenses and criteria based on which I rank the examples also do not have unites.

3. Explain the dash lines in some of your charts? I use dash lines for two purposes. One is to project the example’s position onto the simulation and efficiency axes to show comparisons with other projections. Another is to use dash lines to form closed groups of examples of the same type to draw high-level, inter-group comparisons.

4. Explain the squared shapes of the four sub-areas in some of your charts? The squared shapes of the areas will be seen in later meta-analysis in section 8.2, where I select representative examples from each semantic action and normalize their mappings onto the four areas of the high-level categories of interactions. The four areas are in four rectangular shapes, representing the optimal area where examples achieve both high simulation and efficiency values, the two tension areas where examples achieve one high value and one low value, and the suboptimal area where examples achieve two low values. The areas are rectangular but not in other shapes, such as an oval, because I want them to be parallel to the two axes as a natural segmentation of the whole space. If any of them is an oval, there will be space in the center left that needs further explanation after four ovals are drawn. However, the rectangular sub-areas of the whole space only demonstrate a conceptual division. I do not intend to focus on the strictness of the boundaries.

5. Why use yellow for the simulation axis and blue for the efficiency axis? It is a personal preference. A pure coincidence is that my university – UC Irvine – has blue and yellow as her theme color. I wanted a calm color for efficiency, which is often translated to objective measures such as time, and a warm and bright color for simulation to reinforce the subjective experience. The two colors are also not strictly contrasting to each other, implying that the two values can vary in reconciling the tensions in design.

The following chapters will detail the examples and my analysis. I organize them under broader categories: locomotion interactions, combat interactions, object interactions, and multiplayer and social interactions.

4. Locomotion Interactions

As mentioned before, the locomotion interactions include 1) Move and Turn, 2) Run and Sprint, 3) Teleport, 4) Jump and Crouch, 5) Slide and Glide, 6) Climb, and 7) Special Cases. In the following sections, I will analyze examples of each category using the simulation and efficiency framework.

4.1. Move and Turn

Unfortunately, as the inaugural section, this section is a little boring as it deals with the basic movement and turning actions. The reviewed VR games have overall the same controls and implementations: Alongside physical walking, they all support joystick movement that is by default bound to the left controller. The right joystick is mostly, if not uniformly assigned to the turning action, with options of snap turn, smooth turn, and continuous turn. In addition, players can adjust the move and turn speed in the game options menu. The joystick move and turn controls are arguably the most efficient options. Although they have low interaction fidelity, they do map well onto the direction of moving and turning. The benefit of performing these most commonly performed actions as quickly as possible in the convention of FPS games, in my option, has overweighted the value of simulating their realness.

Beyond these basic settings, in this section, I want to discuss more on additional design features and options related to continuous movement. These features enrich the design space and reveal the value tensions.

4.1.1. Head or Hand Direction

One option tied to the basic continuous movement is whether the joystick control for movement direction is relative to the player's head orientation or hand orientation (which direction the hand is pointing towards). This option has been a long-discussed topic among the online community of VR players and developers, such as this Reddit blog[15] on

---

[15] https://www.reddit.com/r/Vive/comments/ctpepf/do_you_prefer_hand_orientated_movement_or_headset/

the hand-oriented movement and this discussion[16] in the Steam community of the game *Onward* (Downpour Interactive, 2016).

In short, the preference is divided among players, and there are several reasons for each favor. For example, those who prefer the head-based approach explain that they do not need to worry about where their hands are at any time, especially when reloading a gun or swinging dual-wielded weapons. They suggest that the movement can be problematic when the hand switches from the front of the body to the back or side to reach for items. Players who like the hand-based approach mention cases of exploration and shooting, where the gun is mostly in front of the player, making it intuitive and realistic to follow where the weapon is towards. They also mention that it is easy to look around without changing the direction control. On this note, players supporting the head-based movement say they do not have problems compensating the orientation. When they turn to the right while walking straight forward, they can tilt the joystick to the left. In the end, the community have resolved this debate by demanding developers to leave both options as a standard for future games to follow.

My personal preference is the head-oriented movement based on my thoughts on simulation and efficiency. The head-oriented approach is efficient for the familiarity with how movement is implemented in traditional FPS games. It works well for me in the seated mode with the continuous turn. Because the head orientation is where I am looking in the game, it is easier to *see* the forward direction than *think about* where my hand is pointing. In terms of simulation, the two methods are both not of high fidelity. I understand the hand-oriented approach can partially simulate the forward direction of the body when the head is turning to the side, but it is not exactly the body. Still, I think the head-oriented method better simulates the real-world convention of “where I am looking at is the forward direction”.

4.1.2. Pull-based Moving Techniques

Besides the joystick and physical movement, two alternative techniques based on pulling are worth mentioning for their situational uses. First, in the sword fighting game

---

[16] https://steamcommunity.com/app/496240/discussions/0/1681441347883597199/

Ironlights (E McNeill, 2020), players use embodied "pull" or "push" controls to move towards or away from their duel opponent in an arena setting (Figure 44). To initiate the pull action, players first reach their hand and point to a target, at which moment they see a blue line signifying the connection and pulling action. Then they can hold down the "A" button and pull themselves towards the enemy.

For the dueling gameplay, this interaction functions to structure the game dynamics into clear attack and defense phases between characters. The dueling mechanic in this game is less tied to the movement and positioning, than the swinging, blocking, and special abilities with the weapon, which we will see in combat interaction examples. In short, after approaching the enemy, the player can swing the weapon and find gaps between the opponent's defense to attack it. The movement interaction is an efficient design that eases the player to start or end the attack. It becomes more of an effortless commitment to a meaningful action than a skill of movement control to master, allowing the player to focus more on exerting other combat actions.

Figure 44. Pull movement in the sword fighting game *Ironlights*

In terms of simulation, it is of high interaction fidelity and behavioral mimicking of the pulling action. The visual effects on the player's feet during the moving process add to the realness of charging towards the enemy. Therefore, this pull-based movement presents an excellent alternative solution that achieves high values of simulation and efficiency in this specific game context.

Another pull-based movement example in the game *GORN* (Free Lives & 24 Bit Games Pty. Limited, 2017) is to alternate both hands back and forth to pull the world and move the body forward, as illustrated in Figure 45(a). However, this action is cumbersome to perform, although, to some extent, it matches the cumbersome characters in the arena fighting gameplay, as shown in Figures (b) and (c). The characters in this game all have big upper bodies and small legs. They move slowly to smash nearest enemies with various weapons. The pull-based movement gestures seem to fit the narrative that when these big dudes walk around, their strong arms will swing like how the controllers need to swing forward and back.

When the game first introduces the interaction, it is in the scene of a preparing room behind the arena's doors. I felt it was fun to move my body in this way and did not care about how slow it was. The slowness and the rhythm of walking created by arm swing built up my internal emotions and expectations before the battle.

(a)

(b)

(c)

Figure 45. Pull the body to move forward in the game *GORN*

But the first time I showed up to fight in the arena with this movement control, I died in my first fight because I could not quickly dodge the opponent's punch nor move up to hit him as he could do to me. Further, when the enemy characters move around, they do not swing their arms as the control requires. Instead, they always have their fists ready in front of their chests or swing their weapons. Most importantly, when I had to fight these big dudes, I found the fighting gestures, either punching or swinging my weapon, conflict with the pull-based gesture control. I could use one hand to pull to move and the other hand to fight, but when both hands have equipped with weapons or shields, it becomes impossible to move during the fight. Worse still, it is difficult to move backwards with this pull-based control.

Therefore, although this type of pull-based movement may match the overall game aesthetics and the narrative of walking in this big and muscling body, it does not simulate the actual postures in the combat, nor is it efficient for fighting. Compared with the first example in *Ironfight*, this design is less successful as a core game-mechanic option. The only use that I can think of is role-playing, which should be performed in addition to the joystick movement rather than as its alternative. Later soon, we will see examples of running and spriting using this swinging hand technique.

The next two sections discuss visual features that affect the experience of movement interaction.

4.1.3. Tunnel Vision

Many VR games have tunnel vision effects as a comforting option for continuous moving. However, this effect can reveal a tension between simulation and efficiency values. Figure 46 shows four examples representing the problematic design of the tunnel vision effect in VR.

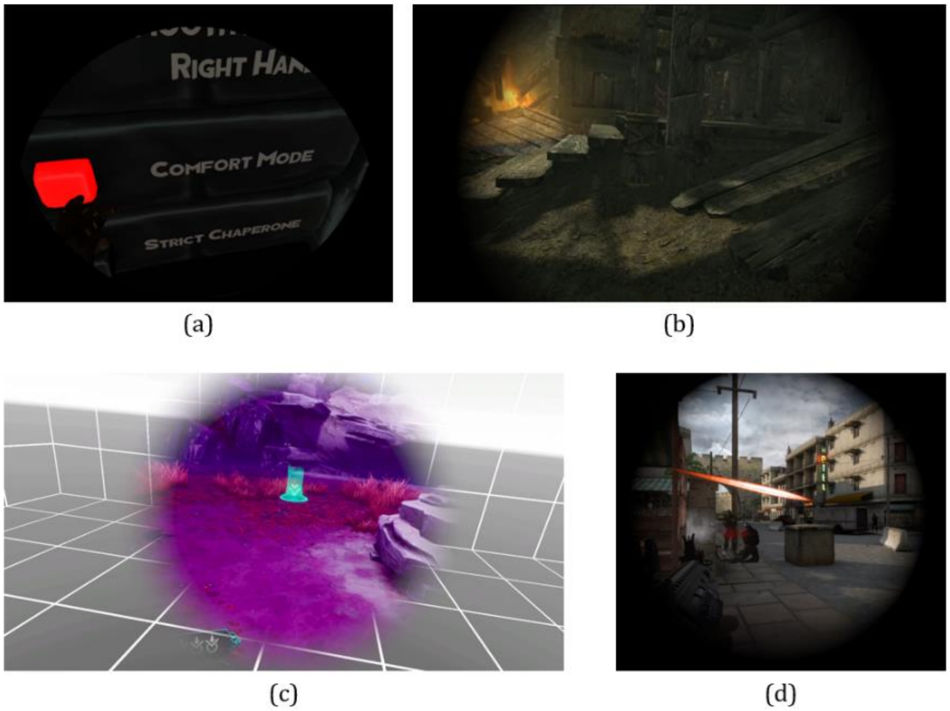


Figure 46. Tunnel vision examples in VR games: (a) *GORN*, (b) *The Elder Scrolls V: Skyrim VR*, (c) *Until You Fall* (Schell Games, 2020), (d) *Onward*

First of all, the design of replacing the outer sight range with a black screen, as shown in examples (a), (b), and (d) or empty virtual space (c), fails to simulate how dynamic vision looks like in the real world. When we move or turn around in the physical world, the sight of the margin gets blurrier, but we do not completely *lose* it. We are still able to have a near 180-degree field of view (FoV). In those examples, however, players are left with even a narrower FoV than the limited 85° supported by the hardware. When I experienced these tunnel vision effects, I felt that I became someone hiding and peeping from a hole instead of being *present* and *immersive* in the virtual world. Besides, these effects have options for different shapes (like an eye shape shown in (a)) and adjustable scopes of tunnel vision. I set the scope to the maximum in example (c) to exaggerate the problem. However, neither the shape nor the scope changes the fact that these are poor simulations of dynamic vision but only serve as the "comforting" option that made me uncomfortable.

Worse still, this design hinders me from seeing incoming enemies or other important scene information in the outer range. The limited vision made me feel that I was losing control of the game world and unable to engage with the content as much as I wanted to. So in my gameplay, I quickly turned off this effect.

Strictly speaking, tunnel vision in example (d) is not triggered through movement but rather through entering the shooting combat. However, I list it here to show that other visual forms, such as the incoming bullet, can still cause uncomfortableness within the tunnelled vision. Furthermore, in the fast-pacing shooting combat gameplay, being able to see the wider scene is crucial. For example, there were enemies on the right side that I did not or was slow to see and turn around due to the black screen. As a result, I could not react immediately and ended up taking the damage out of my control.

In short, these are examples that show how a "failed" visual simulation can also result in inefficiency within the course of interaction. In my opinion, the perfect simulation is to have the sight in the margin be just a little blurred or faded out but still keep the content visible, or not to simulate the dynamic vision at all for the sake of our head remaining still in the physical world.

4.1.4. Collision Handling

When hitting a wall in VR, the player will usually be stopped before penetrating the wall. However, occasionally players can get stuck into a wall or a floor. When such a situation happens, it immediately becomes a disaster that breaks the immersion, forcing players to remove their headsets to stop watching themselves in awkward and uncomfortable spots. Other than getting stuck, players can also feel uncomfortable when running into a wall or facing a close obstacle within a sudden.

The two examples in the following provide special designs for handling such unwelcome collisions.

The first example, shown in Figure 47, allows the player to move *through* obstacles that otherwise will force players to crouch into a potentially uncomfortable place. When passing through the body of obstacles, the visual drastically turns into light green. It highlights the structures of those objects while showing a semi-transparent view of what is in front so players can keep a sense of direction to walk out.

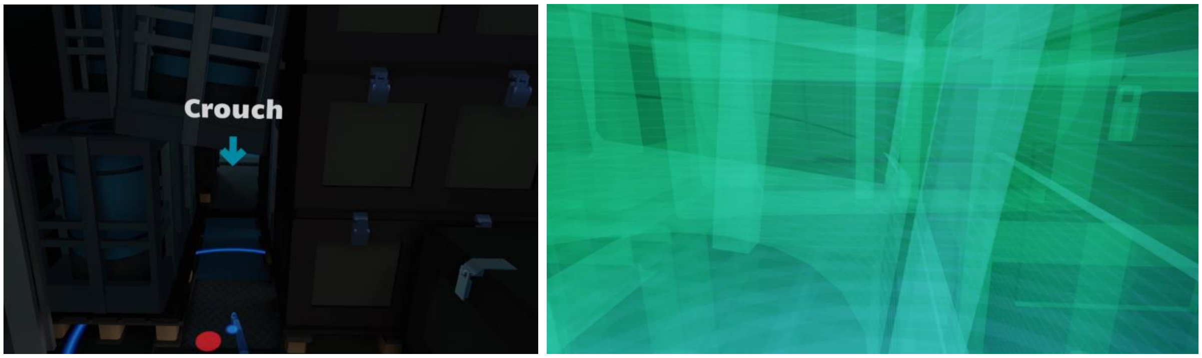


Figure 47. An example of moving through obstacles from *Budget Cuts 2: Mission Insolvency*, an action-adventure stealth game.

Although this design is not an accurate simulation of real-world physics, it gets rationalized by the *stealth* theme of the gameplay. The gameplay is to break through the patrolling of robots by using sneaky movements and attacks. For example, players can preview any spot reachable by their “projectile magnifier” to look around enemy locations before teleporting to that spot. In addition, they can shoot bows and throw weapons to assassinate robot guards and patrols to avoid being detected by them. With these game mechanics, it makes sense for players to have the “cheat option” to go through some obstacles and hang in there. For these reasons, I would argue that this design has a high efficient value for overcoming the gameplay challenges.

However, it is the visual design that reconciles the trade-off between high efficiency over inaccurate simulation. Unlike the tunnel vision examples that completely black out the environmental content, this visual design still simulates the insubstantial and hollow properties of the objects and the sneaky experience of going through them like in a ghost mode by sketching their overall structures.

Another collision handling technique is the black screen transition, as exemplified in *Shadow Legend VR* (VitruviusVR, 2019). When the player’s head is about to face very close to some obstacles during the movement, the screen smoothly and quickly fades into black then backs to normal. Figure 48 shows the transition when I am close to hitting a pillar in a constrained space near the roof ceiling to move around and explore the space on the right.

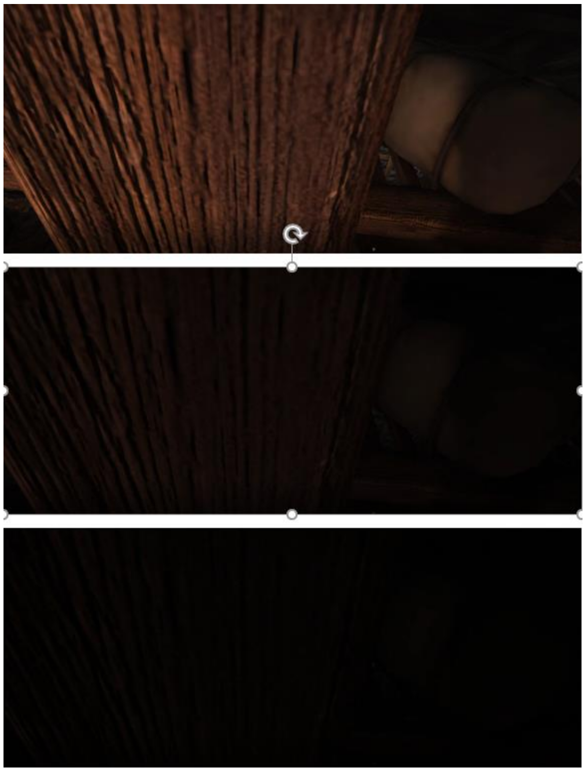


Figure 48. Black screen transition at a pillar close to the ceiling, from *Shadow Legend VR*

Based on my video recording, this transition takes only about one second to complete. In addition, the game system seemed intelligent enough to trigger this effect when I encountered the same spot only for the first one or two times. I interpret this design as a situational protection mechanism that simulates how sometimes I would close my eyes while moving into new scary places, like in an escape room. After I get used to the surroundings, there is no need to protect myself anymore. By the way, if one thinks of the game's name, the overall theming, and artistic style – *shadow* - the black transition is even more realistic and legit.

In conclusion, although this black-screen transition is an extra second in time, it efficiently mitigates potentially uncomfortable movement and keeps the player immersive in the game, and its intelligent first-time detection reconciles potential tensions of simulation and efficiency, making the design achieve relatively high values in both.

Next, we will see design cases of the run and sprint action, a variation of the basic move action with a higher speed but extra control inputs.

4.2. Run and Sprint

Overall the examples present two distinctly different techniques to control run and sprint actions, (1) joystick controls and (2) arm swinging. We will see examples with design details for each technique and evaluate their simulation and efficiency values.

4.2.1. Joystick Controls

The first widely used and perhaps more efficient technique for running and sprinting is to press the left joystick as a *button* while tilting it in any direction. Games such as *Skyrim VR* and *Pavlov VR* (Vankrupt Games, 2017) use this technique. In *Skyrim VR*, there is a stamina bar UI that reduces over time while the player is sprinting. Stamina is a resource consumed not only for sprinting but also for combat interactions such as critical strikes and powerful spells. Like its uses in traditional games, it simulates the character's energy to perform heavy physical actions. In *Pavlov VR*, sprinting is conditional only when you keep your weapon lowered. It is a good simulation of the real-world movement while holding a rifle, in which case one can only run fast by keeping the weapon lowered. Therefore, in terms of simulation, although the joystick-based control is not of high interaction fidelity, these two examples show additional embodied features that add to the realness.

Press the thumbstick to sprint, keep your weapon lowered

Figure 49. Sprint control in *Pavlov VR*

With a nuanced variation in the control, the double-tapping approaches in *Boneworks* (Stress Level Zero, 2019) offer interesting alternative techniques (Figure 50).

Figure 50. Double-tap to run in *Boneworks*

Looking closer at the "joystick as a button" input, I realize that pressing and holding the joystick button for a considerable long time is more challenging than double-tapping it either as a button or to a direction. I think this has to do with the physical design of the joystick. When I press and hold the joystick, I feel a strong counterforce resisting my thumb finger from keeping it down. As we all know, the joystick is not primarily designed as a button. It does not look like a button, nor does it function like the other action buttons on the controller. It is tall, has more resistance, and ergonomically performs poorly as a button.

In contrast, double-tapping the joystick in a direction is more suitable to its affordance and thus easier to perform. The only case where it can slightly be less superior is when the player already starts walking and wants to transition into running. In this case, the player has to move the joystick back to the normal position and then double-taps it, causing the speedy movement to stop for a moment. However, *Boneworks* also allows double-tapping the joystick as a button to run, and it works during the normal speed movement. Therefore, the two solutions using double-tap, in my opinion, present a more efficient solution than the press-and-hold approach.

4.2.2. Arm Swinging

Like the pull-to-move technique for moving, the arm swinging technique requires the player to swing and alternate their two hand controllers back and forth to keep running. This is a much more labor intense and inefficient movement control than the joystick controls. The better simulation it achieves is mimicking the swinging action during the running process. Typically, the faster and more widely you swing, the faster your character will run in the game.

However, I find this technique can be problematic and an over-simulation. Even in the real world, one does not have to swing arms to move faster than the normal walking speed. Only to achieve the maximum speed, such as in a running race or an escaping situation, requires swinging arms fast. However, in the examples that I find, some game design aspects can amplify the problem of this technique.

For example, *STRIDE* (Joy Way, 2020) is a VR parkour-style game where players run, climb, jump between platforms and occasionally have dodge and fight enemy snipers. The arm-swing technique matches the narrative of mastering the motion and body movement techniques. However, I have to acknowledge that it was difficult to have *smooth transitions* between the running action and some other actions in the game. For example, the jump action in this game requires raising both arms high after holding the "A" button. In addition, the game has challenges like wall-running followed by a jump, running into climbing, and running while grabbing the pistol with either hand reaching to its opposite side of the body. These challenges all require reaching out the arm at a tiny time window during running to have the perfect effect. With the arm swinging technique for running, it is technically not easy to have perfect controls and transitions between these transitions.

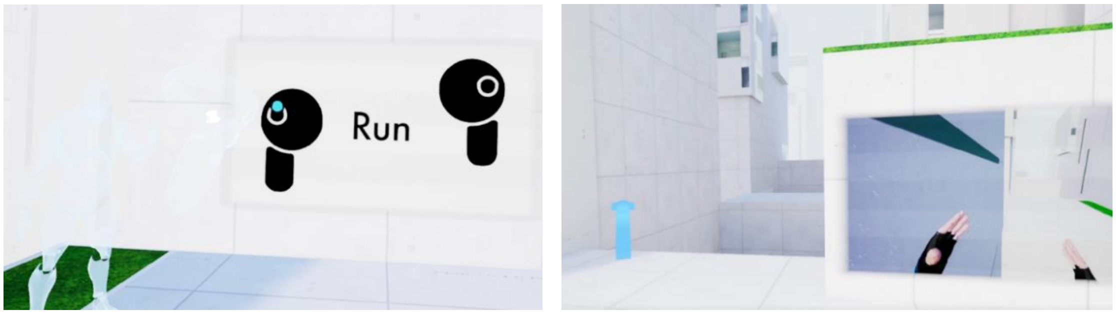


Figure 51. Run action in *STRIDE* and the platformer racing gameplay

In *Blade and Sorcery* (WarpFrog, 2018), the game context of performing the arm swing for running can be different and results in a different experience. This game features sword fighting and magic with similar settings and machines like those in *Skyrim VR*. The arm swinging sprint action can be used in situations like turning away from enemy attacks and running to grab consumables. In this game context, the running becomes more of a short time action when an emergency occurs, rather than an action that needs to be performed for a long time. The game implements other efficient actions such as the force grab to fetch objects from remote that can be used during the running and combat. Still, this arm swing technique in its nature has limitations, particularly when it comes to using it together with other hand-movement-based interactions such as weapon swings.

4.2.3. Summary

To summarize and compare these examples, I put them onto the two-dimensional simulation-efficiency framework (Figure 52) to further evaluate their design values.

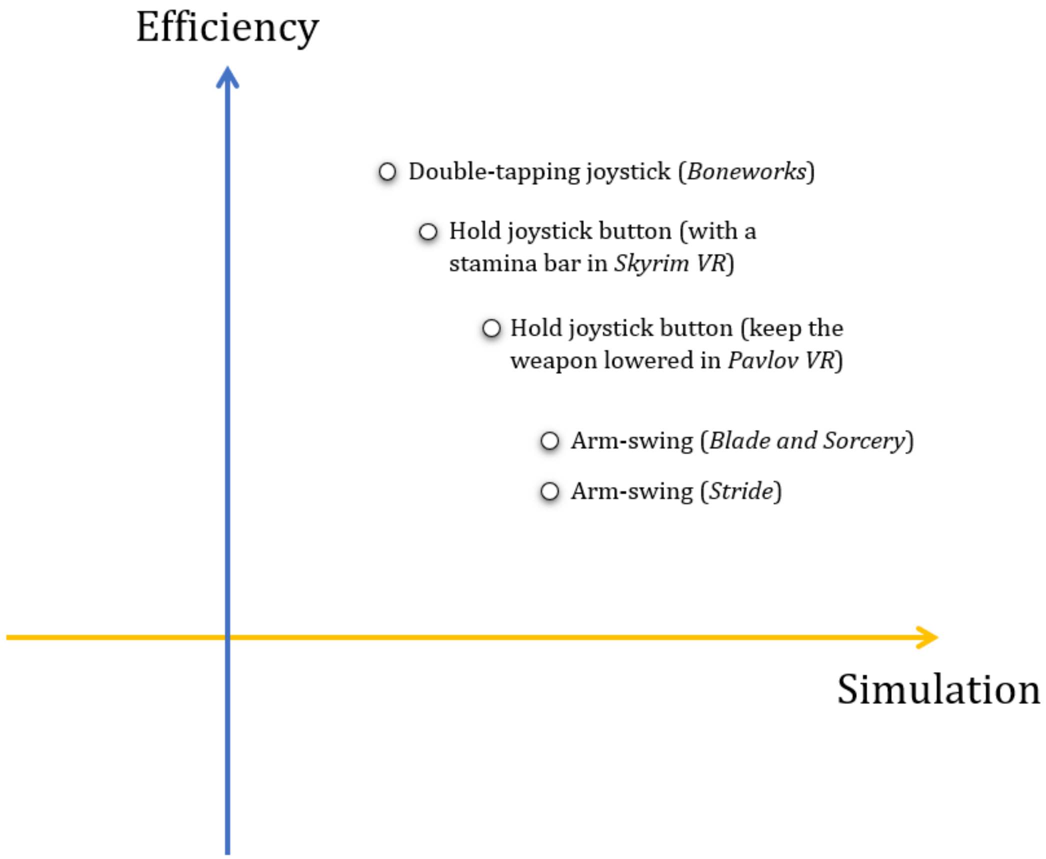


Figure 52. Summary of run and sprint interaction examples

I rank those joystick-based examples to have a higher efficiency value considering their ease of use over the arm swing techniques. Among these joystick controls, I consider double-tapping as the most efficient for the minimal effort it takes with the joystick hardware input. The holding joystick with the weapon lowered approach is of relatively low efficiency because it requires conditional positioning of the arms.

The simulation values within the joystick-based examples are nuanced. The double-tapping can be considered as a metaphor for quickly exerting footstep movements. The control of pressing and holding the joystick button has a low simulation value except exerting more force into the movement control. However, I rank the two examples slightly higher considering the additional metaphorical design of the stamina bar and lowering the weapon.

The two arm-swinging examples partially simulate the physical running postures as swinging hands are not necessary for moving faster. Therefore, I rank them as having a moderate level of simulation. In general, they are less efficient than the joystick controls. The game design in *STRIDE* rewards players who can keep running. Therefore, it amplifies the challenge of performing this action for a long time as a physical skill to master, and thus I rank it to be the less efficient one between the two.

Overall, the run and sprint action reflects a tradeoff between efficiency and simulation. This is because the physical activity of run and sprint is an essentially embodied action, but the VR platform only affords either simple button controls or partially embodied motion controls as the options for game design.

4.3. Teleport

Teleportation is another variation of and, in many cases, an alternative to the basic continuous movement interaction. However, there is no such human behavior called teleportation for games to reference in the real world. In terms of control, most examples use pushing the joystick to the up position to initiate and release to trigger the teleportation, with few using a button press. Therefore, my stimulation analysis of teleportation is mainly based on the interaction's line visuals, additional mechanics, meanings, and game contexts.

4.3.1. Examples

Half-Life: Alex

Half-Life: Alex (Valve Corporation, 2020) provides a comprehensive solution to teleportation with three additional features shown in Figure 53. In my option, this solution achieves both high simulation and efficiency values.

The first feature is how the curve handles the maximum teleportation distance and avoids invalid destinations, such as hitting a wall or falling off an edge. In such situations, the landing part of the curve will turn into a straight line perpendicular to the ground and indicated by two points (Figure 53 - 1). This feature is an elegant solution to show players where the furthest spot they can land in the direction that they are pointing towards. It skips the use of color codes for the line visual to indicate the validity of the destination, as we will see in examples from other games. The straight line is also a good stimulation. It is a metaphor of the teleportation forward course being *cut* by the obstacle (a wall in the picture) or has reached the farthest distance.

Figure 53. Teleportation in *Half-Life: Alyx* and its three additional features: 1) the breaking point where the curve turns into a straight line perpendicular to the ground to indicate the maximum range that the player can teleport; 2) the feet that can be rotated by the left joystick to control the direction the player will be facing after teleportation; 3) the animation of footsteps moving towards the teleport destination.

The second feature is the feet indicator that can be rotated by the left joystick to control where the player will be facing after teleportation (Figure 53 - 2). This is an efficient feature that uses the same joystick to preview and change the orientation quickly without any extra step to change it with the right joystick after landing. In addition, the feet icon provides a good simulation of the orientation change.

The third feature is the animation of footsteps moving from the player to the teleport destination (Figure 53 – 3). This animation only appears after the player holds the joystick in any direction in the initiating phase of teleportation. Nevertheless, it provides a visual simulation of the movement course and eases new VR players into the experience of teleportation.

It is worth noting that *Half-Life: Alyx* also offers an option for players to teleport in a quick blinking by pushing the right-hand joystick down. The blinking works together with the continuous movement – when it is the primary movement for the left joystick. However, in this mode, the feet orientation and animation features are no longer available, and this right-hand teleportation is only intended to blink more quickly than the continuous movement, such as traveling through large empty spaces nonstop. With this option, the overall implementation of teleportation in this game is even more optimized for multiple purposes and application contexts.

Skyrim VR

The next example from *Skyrim VR* uses magic visual effects to tweak more on the curve, with color codes indicating its range and validity. As shown in Figure 54, the line visual applies the game theme about ancient magic power, and the destination is marked with a magic circle. The blue curve shows the close or normal range of teleportation. The green curve shows distant teleportation that will consume stamina, a resource that is coded with the green color in this game to support fast movement and other combat interactions. The green teleportation curve matches the concept of sprinting (which also consumes stamina) discussed in the previous section. Finally, the red curve indicates an invalid teleport destination with an "X" marker.

Figure 54. Teleportation with color-coded line visuals in *Skyrim VR* indicating stamina consumption (green) and invalid destination (red)

The three color-coded line visuals simulate the three respective outcomes and convey the information efficiently. Players immediately know whether they can teleport and whether it will consume their stamina. If they keep seeing the green curve consecutively while traveling in the open world or escaping from enemies, they will know their stamina is running out and have to change their actions accordingly.

Sairento VR

Sairento VR (Mixed Realms & Swag Soft Holdings Pte Ltd, 2016) blends the teleport interaction with its theme of cyber ninja combat adventure. Like the control scheme in *Alyx*, this teleportation by default can be performed alongside continuous movement. It can be performed by clicking the left joystick button or pushing the right joystick in the up direction. If players have experienced the quick right-hand teleportation in *Half-Life: Alyx* and the joystick-based sprinting in *Skyrim VR*, they will get familiar with the versatile controls in this game.

The visual forms, the blue "stamina" bar that also slows the time, and additional mechanics such as double-jumping and wall-gliding make this teleportation example even more unique. The visual forms consist of the white curve for quick dashes, the blue curve for the jump, and the triangles forming the curve to match the game's cyber theme. A simple and clear light spot indicates the landing position, and during consecutive quick dashes, the curve will disappear, leaving only the white light spot for visual simplicity.

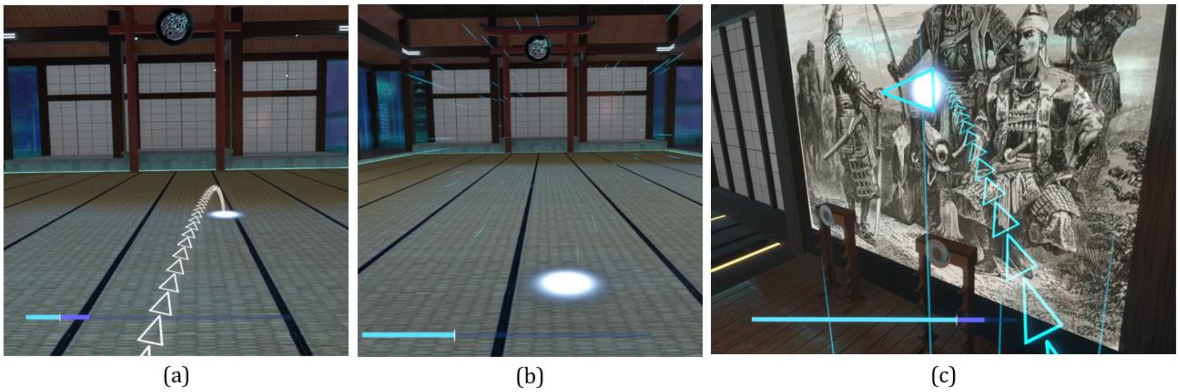


Figure 55. Teleportation and visuals in *Sairento VR*; (a) the white curve indicating a quick forward dash; (b) the curve disappears, leaving only the light spot and airflow; (c) the blue curve indicating jumping, which is a variation of teleporting in this game

When the player holds the teleportation curve, the blue bar at the bottom of the screen shows up, showing the energy left for extra teleportation. Also, the game slows the time during the "holding to teleport" phase, buying time in combat for players to think about a destination without being hit by enemies while thinking. Overall, the energy bar recharges quickly and supports a strong cyber ninja movement style and a fast pace of gameplay. During the teleportation, the screen shows airflows to simulate the dashing. Overall, I think this design achieves high simulation and efficiency values. It also fits well into the game context and leverages the platform affordance for multiple inputs and mechanics, which we will revisit in later sections.

Budget Cuts 2

The teleportation in *Budget Cuts 2: Mission Insolvency* (Neat Corporation & Fast Travel Games, 2019), or *Budget Cuts 2*, has an option that uniquely mixes with the stealth game mechanics. To operate what is shown in Figure 56, players first select either hand to equip the translocator, which can shoot a blue projectile ball by pressing the trigger button. After the projectile lands, players get its sight and can rotate to preview its surroundings before pressing the trigger again to teleport to that location.

Figure 56. The teleportation process in *Budge Cuts 2*, featuring shooting a bounceable projectile and the ability to preview the destination before teleporting

This technique can be operated by either hand pressing the trigger button. It also works alongside the joystick-based continuous movement. From the game design perspective, it combines the fast-traveling aspect with the mechanic of gaining sight in the distance to better sneak and survive in the game.

Another interesting mechanic is that the projectile ball shot from the translocator can bounce between objects before it finally loses momentum and stops. Thus, this mechanic can skip some jumping and crouching to reach places out of the player's line of sight. It also clearly visualizes the trajectory as if the player would follow it. The only inefficiency is that players have to wait until the ball stops to preview and teleport to the new location.

Waltz of the Wizard: Natural Magic

Unlike the above cases the each shows a unique design that realizes high simulation and efficiency values, the following example shows a problematic and inefficient teleportation technique.

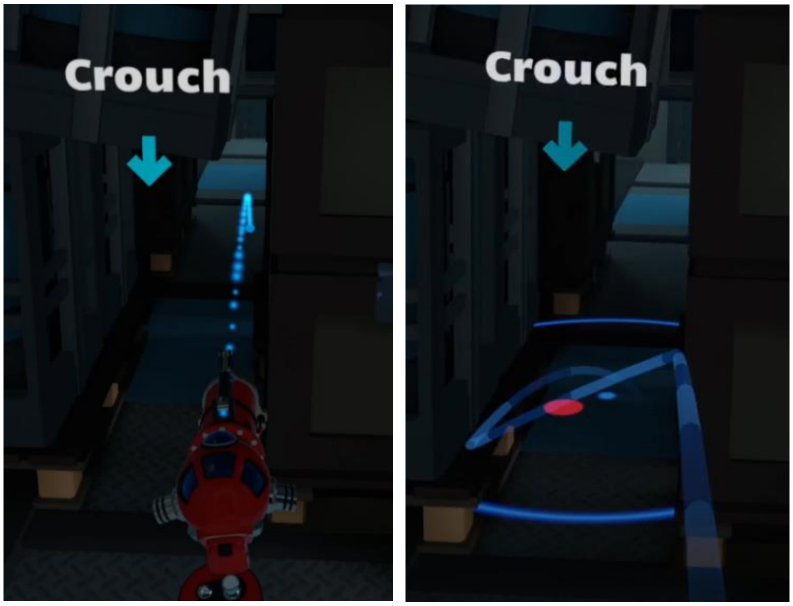


Figure 57. The teleportation technique in *Budget Cuts 2* can be used to skip crouching, but its bouncing mechanic can be inefficient and land to invalid spot

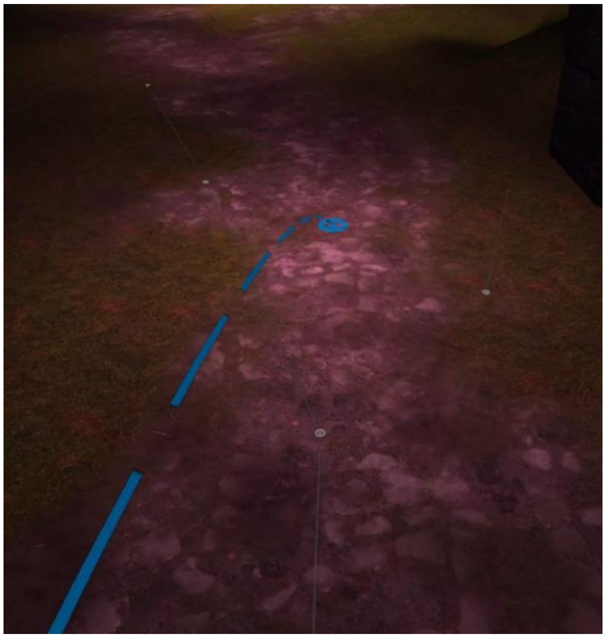


Figure 58. An inefficient "zig-zagging" teleportation example from *Waltz of the Wizard: Natural Magic* (Aldin Dynamics, 2021)

As shown in Figure 58, this technique is interesting in that it sets a breakpoint each time after the reticle points to a spot and hovers on it for a short amount of time. Then the player can continue to point at a new location and generate another breakpoint. Once the player releases the joystick, the character will traverse through all the breakpoints, following the lines connecting between them. The problem, however, is that when the player reaches out far then changes the mind to aim for a closer spot, the movement will still go to the further spot then move backwards. This zig-zagging movement can be a frustrating experience for any player. Also, the teleportation curve visual and the overall design barely connect to the game's wizard theme, while other playful content in this game has done so well.

4.3.2. Summary

As shown in Figure 59, I map the selected teleportation examples onto the simulation and efficiency framework and summarize my reasoning of their values for the comparison. Overall, these examples have moderate to high levels of simulation and efficiency, except the *Waltz of the Wizard* example is low in both.

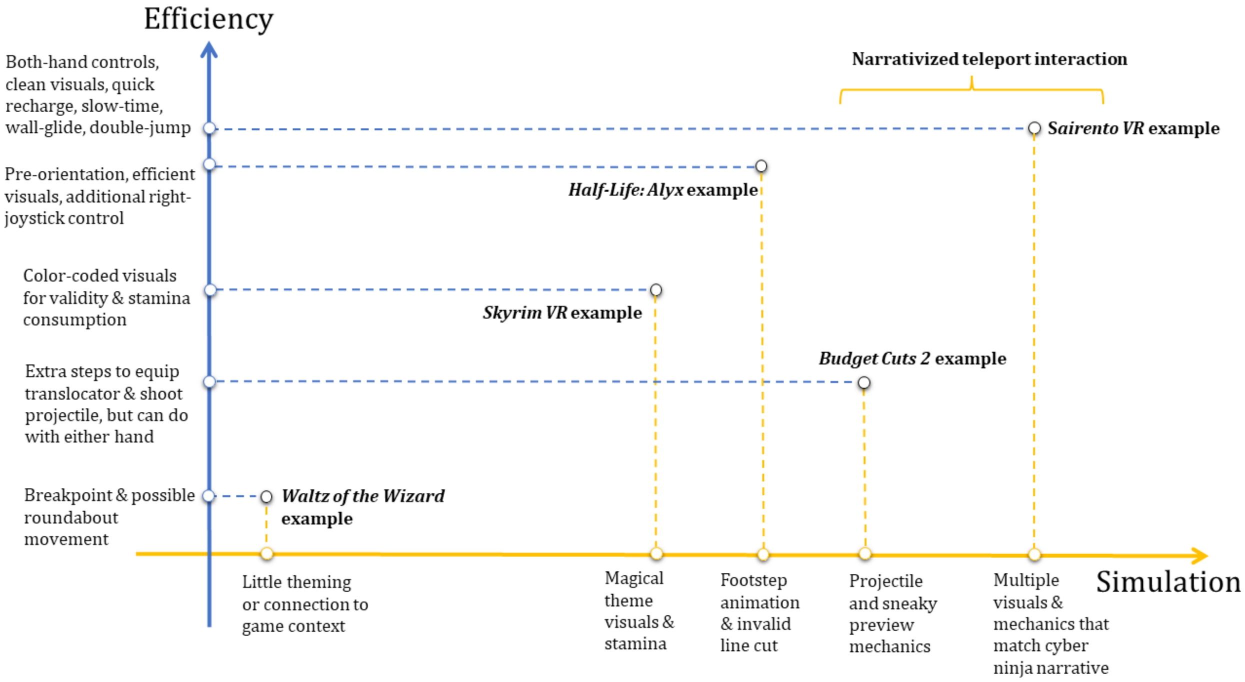


Figure 59. Summary of teleport interaction examples

Because teleportation is a hyperreality action and already has a standard movement form, its simulation value mainly comes from visuals and how well the interaction fits the overall game theme and context.

The two examples from *Budget Cuts 2* and *Sairento VR* stand out in this regard. When I performed these two examples of teleportation, I found they were not just a move action. They became new game *mechanics* bearing additional narrative meanings in the game context. The *Budget Cuts 2* teleportation became a way to gain sight in the distance, benefiting stealth gameplay. The *Sairento VR* made me feel dashing, double jumping, and wall running in a cyber ninja body. These two "narrativized teleport interaction" examples reflect the functional narrativization of game interfaces (Tanenbaum & Bizzocchi, 2009). They go beyond the form of teleportation to simulate meaningful gameplay mechanics and narratives. Also, they show the efficiency of coupling multiple actions into the same form and control. Considering other usability-related factors, I find the *Sairento VR* have the highest efficiency value among all the examples. It can be performed by either hand, has a clean visual where the line can disappear during quick dashes so as not to be visually overwhelming. Its quick-recharging and slow-time mechanics can save players a lot of time in combat. The *Budget Cuts 2* example is relatively less efficient in terms of the steps for performing the action because it requires equipping the translocator and waiting for the projectile to land. However, it can be performed using either hand.

The *Half-Life: Alyx* and *Skyrim VR* examples do not have strong narrativization, but they still present moderate to high levels of simulation and efficiency for the teleport action per se. The *Skyrim VR* example uses magic and color as metaphors. The *Half-Life: Alyx* implementation supports the pre-orientation mechanic and option to perform a quick forward shift with the right-joystick in addition to the left-joystick-based movement, making it overall a highly efficient locomotion design.

4.4. Jump and Crouch

The jump and crouch interaction examples can be divided by the techniques (1) physical height and actions, (2) button-and-joystick-based, and (3) teleport-based. These two techniques differ obviously in simulation and efficiency values and trade one for another. Still, some examples implement a hybrid of the two techniques and involve other elements that amplify or reconcile the tensions between the design values.

4.4.1. Physical Height and Actions

To start with, thanks to the VR head position tracking, players can always adjust the height of their characters by physically standing up straight, sitting down in a chair, or further lowering their bodies. In some games, the crouch action is uniquely controlled by physically crouching, such as in the *Budge Cuts 2* example that we saw above. This is absolutely a high simulation for the crouch action. However, it generally requires the player to be gaming in a standing mode, and the physical crouching itself can be inefficient to perform. In the seated mode, it is unable to physically crouch. The same is true regarding physically jumping, which is more rarely seen as a core game mechanic than the physical crouching in VR games. In some situations, players can jump physically to grab items such as a coin located on the beam near the ceiling. Still, the physical jump is more of a height adjustment rather than a substantial mechanic.

The two representative examples that we will see either makes the physical crouch an occasional and situational challenge in the game or a bonus action to perform. In the parkour racing game *STRIDE*, players have to crouch to glide through the underneath of an obstacle while running towards it (Figure 60). Otherwise, they will be hit and slowed in the race. This challenge is like the one in Figure 57 from *Budget Cuts 2,* but there is no cheat option to avoid physically crouching. However, when I played this game challenge, I found that I did not have to perform a deep crouch physically but only slightly bend my legs to lower my head. Therefore, it is not highly inefficient.

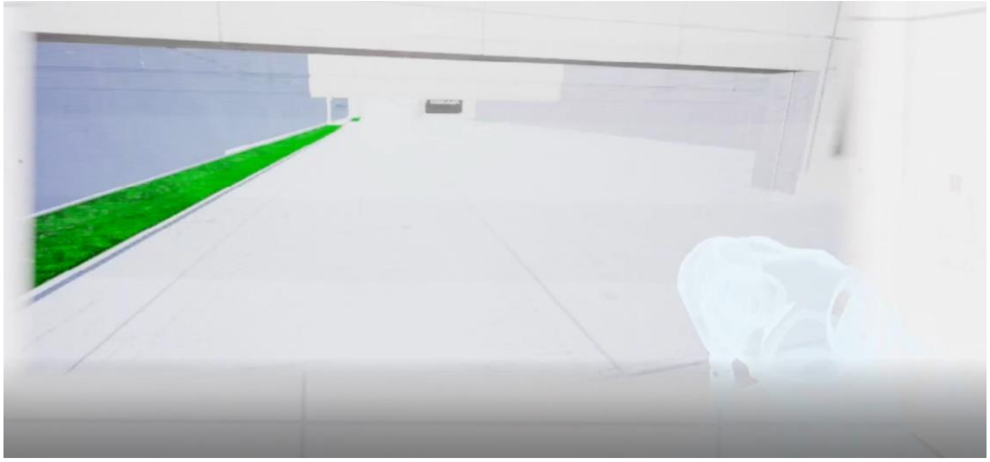


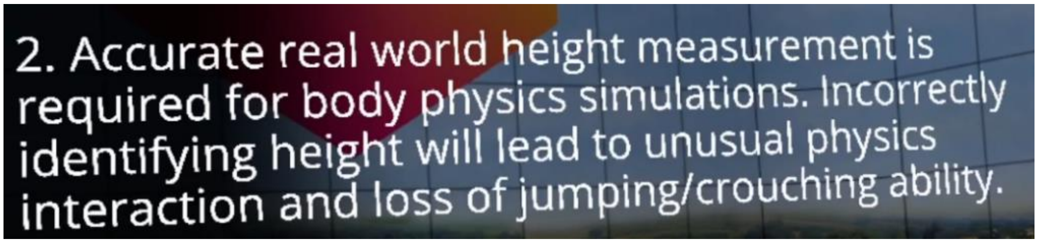
Figure 60. A “Crouch” or lower height challenge to glide through the underneath of an obstacle, *STRIDE*

Another example from *Boneworks* uses crouching before jumping to simulate the initiation of a “boosted” jump. However, the jump action is performed through a joystick button. To perform a boosted jump, one has to combine four actions together: (1) crouch by lowering hands and head, (2) use the special slow-time mechanic by pressing another button, (3) hold before releasing the jump (“A”) button to jump, (4) add room-scale input by standing up and raising the two hands high when jumping, additionally, they can physically rotate their arms in the air and crouch again before landing the jump for maximum results.

This whole sequence simulates the charging process of the burst jump power and the transferring of upward momentum like how a big jump is done in the real world. It rewards the player with three times the height of a standard jump. The mechanic is an outcome of the game’s physics simulation and can be efficient in some game levels. However, the game itself is more of a tech demo, and the actions require a lot of practice and skills to master.

Figure 61 Boosted jump in *Boneworks*, screenshot from a YouTube gameplay video[17]: https://www.youtube.com/watch?v=wuJdl02rrX8

In addition, many of these hardcore physics simulation games featuring embodied jump and crouch actions require a height calibration and playing in the standing mode for all the physics to work perfectly. As a result, they typically require the player to accurately and manually enter and adjust their real-world height at the start of the game.



2. Accurate real world height measurement is required for body physics simulations. Incorrectly identifying height will lead to unusual physics interaction and loss of jumping/crouching ability.

Figure 62. Height measurement required in *Boneworks*

A disadvantage of physical height adjustment that is more related to game challenges and combat is when players encounter enemies or in an urgent situation but have to reach the floor with their hands to grab something. In some cases, a player can accidentally drop the weapon, and because crouch physically takes time and lowers the head, it can force the player in a vulnerable and surrendering position to an incoming attack while they are trying to pick up their weapon. Some games implement extra functionalities to avoid this awkward experience, such as controller-based crouching,

---

[17] The video titled "How to JUMP PROPERLY in BONEWORKS | 20ft 'Ninja Jump' guide" has over 20k views at the time of writing this dissertation and suggests that the boosted jump action is complicated to many players.

remote grabbing, and even preventing dropping the weapon. We will further discuss these topics in later sections.

4.4.2. Button-and-Joystick-based

The button-press or joystick-based controls for crouching and jumping are more commonly used in addition to the physical height adjustment. It improves the efficiency of gameplay in either seated or stand mode. For example, in the task of collecting runes in the game *Shadow Legend VR*, I could press the “A” button in seated mode to lower my character. As shown in Figure 63’s situation, I was on a horizontal ladder trying to grab the green rune on a roof. Had I chosen to lower my body physically, I would feel that I was in danger of falling off and could only slowly move my body.

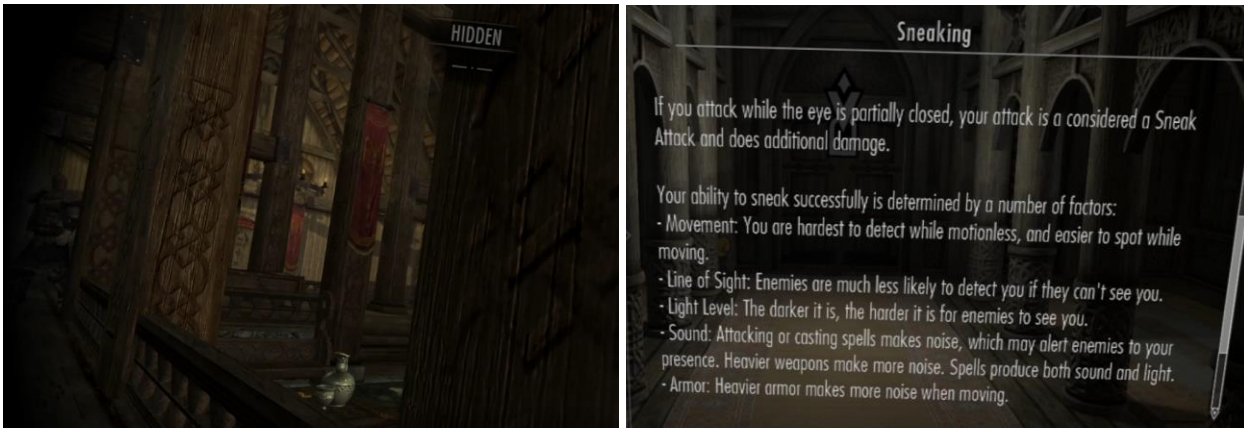
Figure 63. Discrete height lowering or “crouch” to easily grab the rune in *Shadow Legend VR*

In other games, the “A” button is assigned to jump rather than crouch. Some perform the jump action by simply pressing the button, while others trigger the jump by releasing it to use the holding as a charging phase. Acknowledgedly, the disembodied jump can cause discomfort for some players and make them visually unprepared to see the sudden vertical upraising while their real-world bodies are still. In the very first scene of *Blade and Sorcery*, the game, in my opinion, deliberately shows a mirror (Figure 64) through which the player can see their character and the jump action to reconcile a bit for the disconnection.

Figure 64. In the very first scene, *Blade and Sorcery* allows players to see through a mirror the button-press jump action

Skyrim VR also features this type of jumping control, although it is assigned to the right joystick pushing up because the "A" button is used as a general action button for search, take, talk, steal, etc. The jump action in *Skyrim VR* is used for going uphills and moving across small obstacles as it only lifts the character by a normal height. In terms of simulation and efficiency, the joystick-push and button-press do not have a significant difference. The joystick method matches the conceptual direction of jump "up", while the button-press method reinforces the immediate burst of jumping, but they both loosely connect to simulation, and their execution time is almost equal.

When viewing *Skyrim VR*'s locomotion control scheme as a whole and considering the symmetric implementation of the crouch action on the joystick down, it makes more sense to credit extra simulation value for the consistent mapping of jump and crouch in opposite directions of the right joystick. In addition, the crouch action in *Skyrim VR*'s game context also functions as a stealth mechanic, which lowers nearby enemies and guards' attention to the player.



Sneaking

If you attack while the eye is partially closed, your attack is a considered a Sneak Attack and does additional damage.

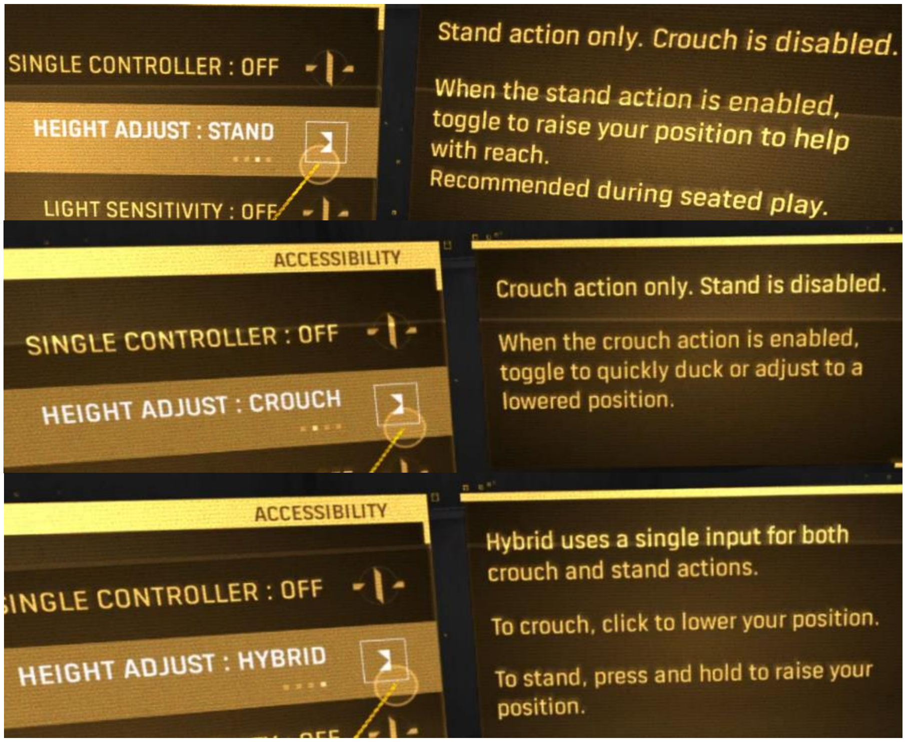
Your ability to sneak successfully is determined by a number of factors:
- Movement: You are hardest to detect while motionless, and easier to spot while moving.
- Line of Sight: Enemies are much less likely to detect you if they can't see you.
- Light Level: The darker it is, the harder it is for enemies to see you.
- Sound: Attacking or casting spells makes noise, which may alert enemies to your presence. Heavier weapons make more noise. Spells produce both sound and light.
- Armor: Heavier armor makes more noise when moving.

Figure 65. Crouching in *Skyrim VR* enters the stealth mode that lowers the player's movement speed and nearby people's attention to the player.

The jumping example from *STRIDE* involves a mix of embodied and button-press inputs. As demonstrated in Figure 66, players hold the "A" button first, then reach out both hands while releasing the button to jump. Similar to my critiques of its arm-swinging running action, this jump action only partially simulates the physical behavior and can be error-prone when trying to activate it in the transition from running. It also encourages holding the button longer and reaching out farther to perform bigger jumps. Jumping in this game is a highly frequent action, and in my experience, the activities would make me sweat even after a short run.

Figure 66. The jump interaction in *STRIDE* includes pushing the jump button and releasing it with both hands reaching out

Last but not least, one key point that I want to highlight with the following *Half-Life: Alyx* example is the difference among "height adjust", "jump", and "crouch". This FPS game provides three options for height adjustments that translate to stand and crouch using the left joystick.



SINGLE CONTROLLER : OFF

HEIGHT ADJUST : STAND

LIGHT SENSITIVITY : OFF

Stand action only. Crouch is disabled.

When the stand action is enabled, toggle to raise your position to help with reach.
Recommended during seated play.

ACCESSIBILITY

SINGLE CONTROLLER : OFF

HEIGHT ADJUST : CROUCH

Crouch action only. Stand is disabled.

When the crouch action is enabled, toggle to quickly duck or adjust to a lowered position.

ACCESSIBILITY

INGLE CONTROLLER : OFF

HEIGHT ADJUST : HYBRID

Hybrid uses a single input for both crouch and stand actions.

To crouch, click to lower your position.

To stand, press and hold to raise your position.

Figure 67. Three options in *Half-Life: Alyx* for height adjust and controls: stand, crouch, and hybrid

These three options present an inclusive solution to support both standing and seated gaming and player preferences. The idea is first to establish a preset play style and then choose the best options for the player. However, it takes some time to try out these options as their descriptions can be misleading. For example, the first (stand) option is recommended during seated play. However, the crouch option is disabled in this mode, which can be inefficient in many situations, such as grabbing. This is echoed in an online discussion in the Steam community, where players have argued that “seated gaming absolutely requires a crouch functionality[18].” The collective wisdom of the players in the discussion reveals the hybrid mode and the crouch mode more suitable for seated play. In the hybrid mode, the player can shift between three height levels. A user suggests that by

---

[18] The discussion starts with a usability complaint about the setting the height in game. It also mentions how confusing the “stand” naming is for seated play, see https://steamcommunity.com/app/546560/discussions/0/2137462524934031899

default, the player can stay in the standing height by clicking and holding the left joystick, click to lower to a middle crouch, and click again to a lower crouch.

From the efficiency perspective, I prefer the crouch mode that only toggles the crouch over the hybrid mode that involves too many levels. In the seated mode, I can still set the character's standing height to be my normal height in the game and simply click to crouch to reach the lower ground.

As one user says in the discussion, "… because if VR was the future of gaming, it probably wouldn't stay that if you always had to stand to play something." As a triple-A game, *Half-Life: Alyx*'s playtime can easily exceed over ten hours. For long-time gaming, standing can quickly lead to fatigue. Providing efficient interactions for seated play, such as the height adjustment and crouch functionality, is considered more and more important in VR games. Another user mentions a friend using a wheelchair and emphasizes the necessity to support seated play for accessibility and inclusion.

Therefore, in this case, the inaccurate simulation of the physical height in the game becomes a less priority than the value of efficiency and accessibility in supporting the crouch action in the seated play, which is an underestimated factor of the VR platform in VR game design practices.

4.4.3. Teleportation-based

Half-Life: Alyx's jump example transitions our discussion from button-and-joystick-based techniques to teleportation-based jump interaction. As shown in Figure 68, the "jump indicator" is essentially a narrativization of its teleport interaction. Developers reuse the same visual asset, including the line visual and the feet icon, and the control techniques of bringing the joystick up then releasing it. The difference is that the feet icon does not support rotation and pre-orientation because doing it with the right joystick can conflict with the turning action controls.

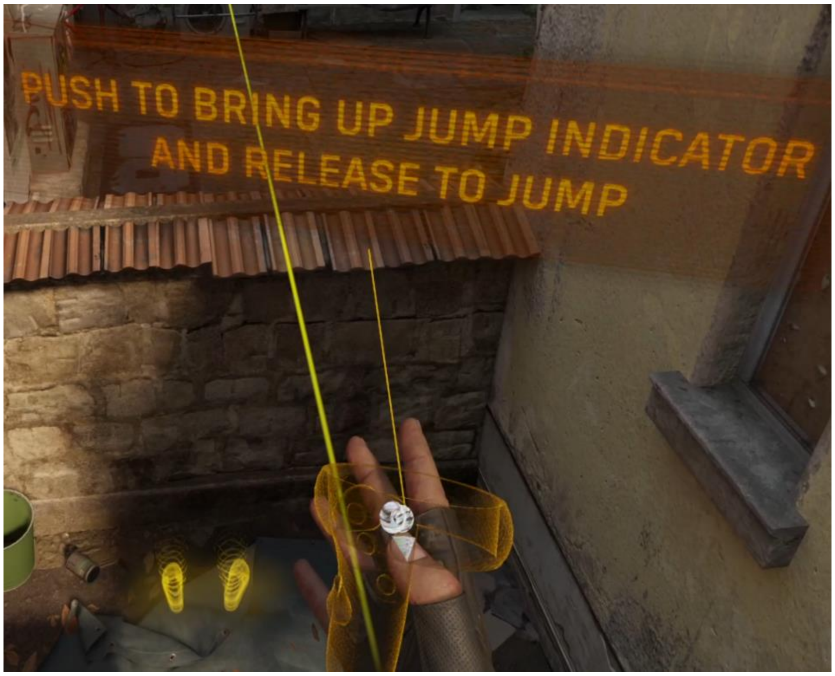


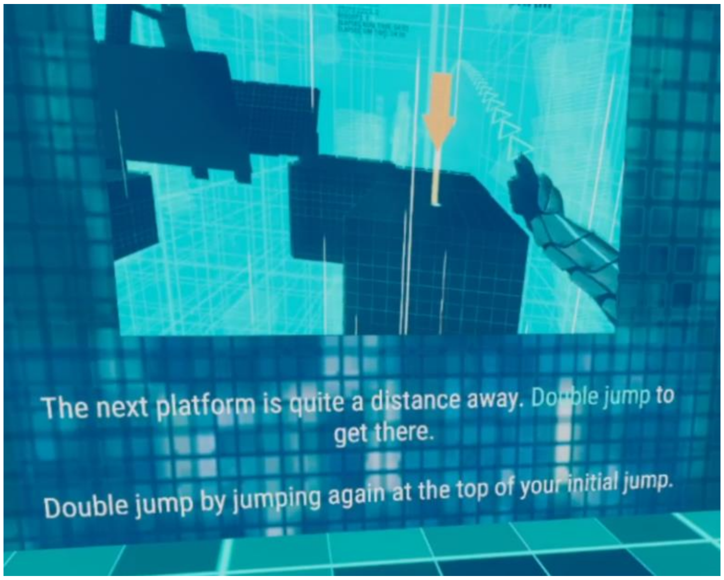
Figure 68. *Half-Life Alyx*'s jump indicator, which is essentially a narrativization of its teleport interaction

While the crouch action is technically the same as lower the height, the jump is not as increasing the height, but rather getting across a gap to reach another platform in the distance. Rather than assign a new button, the reuse of the teleportation is an efficient solution given the VR platform's constraints and the richness of actions to assign controls to in this game.

Furthermore, when there is a short obstacle in front of the player, the "jump" technique of holding the right joystick forward turns into what the developers call the "mantle" action, which quickly climbs over the obstacle. This is a case where the same input technique is used for multiple semantic actions in the game design.

When combining the mantling and the teleportation-based jumping, the overall effect is arguably more efficient than using the same jump action, like the one in *Skyrim VR*, to get over the long gap and the shorter obstacle. They also show more realistic behaviors and animation tailored to each case.

Another teleportation-based jump example is the cyber ninja jumps that we are revisiting in *Sairento VR*. The beauty of this game's jumping mechanics is that you can visualize the trajectory and the landing spot of the jump, you can land on any surface, including on the wall, and you can double jump by jumping again at the top of your initial jump (Figure 69). You will hear the sound mimicking hitting the air as you jump.



The next platform is quite a distance away. Double jump to get there.

Double jump by jumping again at the top of your initial jump.

Figure 69. Double jump in *Sairento VR*

With these mechanics combined, the jump interactions become powerful, efficient and fun activities to play with. They also benefit the gameplay. For example, you can reach a distant platform and bounce against the wall to avoid falling and make sure to land on the platform as long as you can point the curve onto the ground, as demonstrated in Figure 70. In addition, the stamina recharges quickly upon landing on any surface, giving players a lot of room to perform the jump controls.

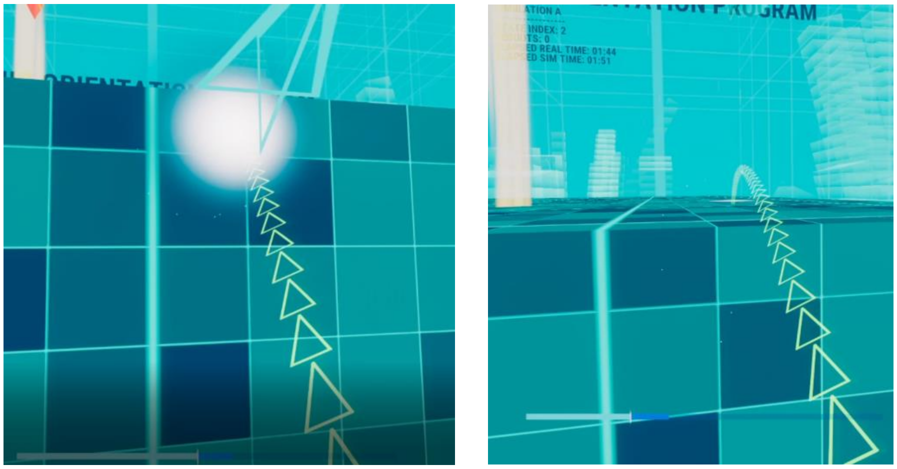


Figure 70. Double jump in the air bouncing from the edge to land on the platform without falling off, in *Sairento VR*

Finally, a remarkable variation of the jump in mid-air is to open both hands to flip the body upside down, adding to the playful ninja-themed movements and extending the possibility of combat interactions, such as shooting enemies from the ceiling. User comments on a tutorial video[19] of this interaction mention that although it is an additional feature and seems to be difficult for some players, it simulates the flipping action by pretending that "you are really gonna flip and put your arms up in the air when you jump (or at the peak of your jump) up and to the side for side flips." Overall, the jump interaction in this game has achieved high values of simulation and efficiency. They also offer different levels of detail and variations to suit players' preferences and skills.

4.4.4. Summary

In summary, the crouch interaction examples are divided into 1) the controller-based simple inputs that adjust the height between discrete levels and 2) the physical and analog height adjustment. While the trade-offs between simulation and efficiency are also obvious for these two approaches, the platform factor of seated play amplifies the tension. The VR community resolves it by prioritizing the efficient discrete crouch functionality over the physical crouch to guarantee the accessibility and inclusion of the seated play mode.

The design space of VR jump interaction is more diversified. Once again, I use the simulation and efficiency frame to map the representative examples for clear visualization and analysis. The results are shown in Figure 71.

The platform constraints still significantly affect the mimicking approaches of the jump actions. The two examples from *STRIDE* and *Boneworks* encourage and reward the standing playstyle; however, due to the nature of the jumping activity, they only partially mimic the hand actions of reaching out but abstract the jump action into holding the button then releasing it. *Boneworks* scores relatively higher in simulation because its boosted jump features a lot of physics and embodied actions. However, the room-scale input of crouching

---

[19] Sairento VR - How to do flips (Tutorial), made by Youtube user Lord Taryn: https://www.youtube.com/watch?v=Pau1CHxtrWU

and additional hand movements in *Boneworks* only add to the niche case of performing a maximum boosted jump, which can be a time-saving reward for skills and practice.

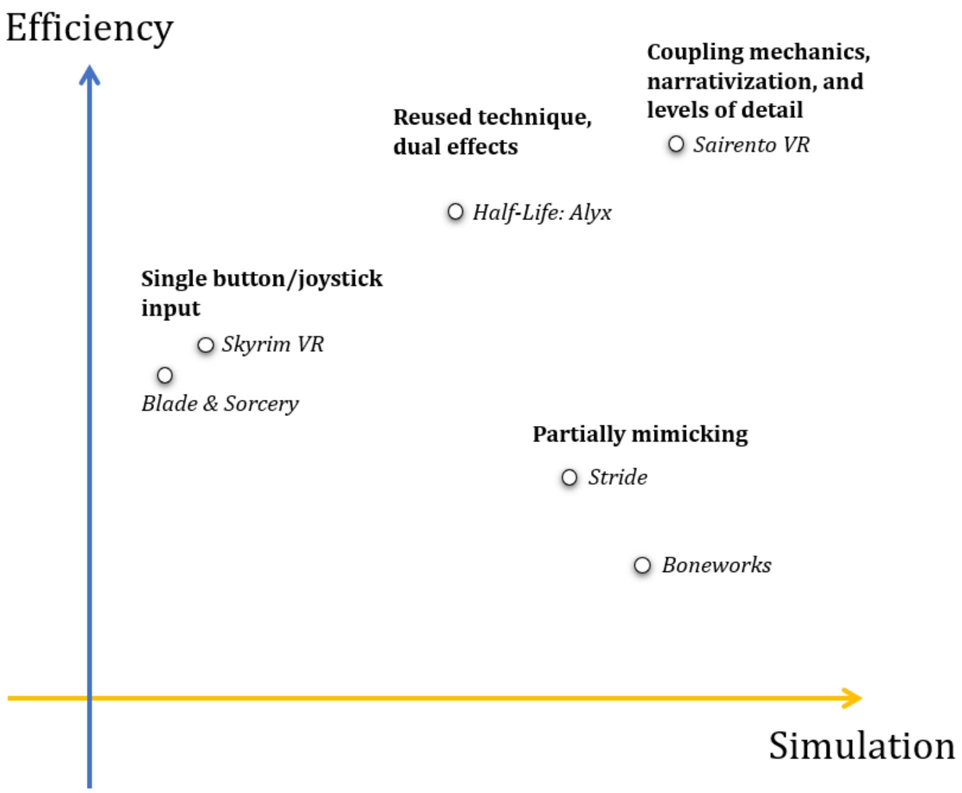


Figure 71. Jump interaction design cases: summary and comparison

Among the more efficient examples built on relatively simple controls, the roles of game context and design aspects become more salient. For the jump action per se, the button press and joystick push are of equal efficiency. I give the *Skyrim VR* implement slightly more in both to reveal the context that it supports both jump and crouch symmetrically by mapping them in opposite joystick directions and that the crouching also triggers sneaky mechanics.

The two outstanding examples from *Half-Life: Alyx* and *Sairento VR* how game design elements can amplify each value and reconcile potential tensions. Both feature reuse of input techniques but achieve greater combined effects. The *Alyx* example multiplexes the pushing right joystick up control to suit longer jumping across a gap and smaller mantling to quickly get over obstacles. Its diversified semantics and tailored visuals compensate for its abstract controller-based interaction. Reusing the visual assets for teleportation, it narrativizes the feet graphics into the jump indicator.

Sairento VR's jump interaction is overall optimal in terms of both design values and well suited to its game context. It beautifully blends multiple locomotion mechanics and provides levels of detail and variations. Its double-jumping, wall hovering, recharging, slow-time, and sliding and flipping bring a lot of efficiency value in the gameplay (e.g. hovering the wall repeatedly to reach a high ground). Also, these interactions work well with the game's unique cyber ninja style. Although they do not excel in interaction fidelity or behavioral mimicking, their narrativization and theming with mechanics and visuals make them feel more immersive and realistic than the partially mimicking controls.

4.5. Slide and Glide

These two actions, slide and glide, are attractive options using momentum and inertia to keep moving or flying forward. Overall they are less commonly implemented than other locomotion techniques but add to the character's mobility and interactivity. Although there are fewer examples to compare in this category, I feel it is worth studying each example's unique design and techniques and how they incorporate basic movements.

4.5.1. Crouch-based Slide, Wall-Run, and Zipline

The crouch-based technique of sliding presents a good simulation for the physical action. Again, in *Sairento VR*, a game that features many locomotion techniques, players can physically crouch before landing from a jump of any height to slide along the ground. Of course, the angle between the landing momentum and the surface needs to be acute to activate the sliding, like how its physics works in the real world. As a counterexample, I tried to jump on a wall with my face hitting it and could not trigger the slide no matter how hard I crouched.

Figure 72. Slide and wall-run in *Sairento VR*

Figure 72 shows a gameplay example of sliding on a tilted wall to achieve a wall-running effect. As we can see in the left picture, the visual design of the game's jump curve shows an arrow on the landing location to signal the player the direction that they will continue sliding along. It is an efficient design in this regard. Alternatively, players can chain their jumps when landing on a wall for the wall running effect. *Sairento VR* also supports the player to slide after a quick dash (teleport). When combined with a sword, the player can cut through enemies during the slide (Figure 73).

Figure 73. Slide with a sword to cut through enemies during the sliding, *Sairento VR,* screenshots from an online gameplay video (8:40): https://www.youtube.com/watch?v=Mt5YxdmoX40

As we mentioned before, the same crouch-based sliding technique is used in *STRIDE,* but the sliding only activates during running for challenges like passing through an obstacle underneath. *STRIDE* also enables wall-running, but it requires running after jumping on the wall rather than sliding as a simulation of real-world parkour physics. Figure 74 shows a gameplay example of wall running where the player has to keep swinging hands after jumping onto the wall.

Figure 74. Wall running in *STRIDE* requires jumping onto the wall then running with the hand swinging technique.

Before leaving the discussion on crouch-based sliding, I want to focus more on the potential inefficient part within the technique, which is that it requires physical crouching. However, as discussed before, physical crouching is essentially height adjustment and is not recommended for seated play. That means players who prefer sitting to play these VR games can hardly play with the sliding option that gives them increased mobility or even is a must-do for passing some challenge.

However, even with the standing play style, the physical crouch can also be problematic regarding the threshold for switching between the standing and crouching states. Because the sliding uses crouching, or more preciously, a lower height threshold value as a state *trigger*, for players, the control essentially becomes lowering their head to pass below that threshold number set by developers. In my experience, it can be difficult to set that perfect threshold value for all players. If the threshold is set too high, for example, 0.9 of the player’s normal height, it will lead to problems like when players accidentally enter the crouch state and trigger sliding that actually, they do not want. Some people may just want to adjust their height during the jump or see something in the game by lowering the head slightly. I mention this because I experienced a false positive of sliding in *Sairento VR* when reaching a wall. However, if the threshold is set too low, for example, 0.6 of the player's normal height, it will make the action cumbersome to perform. In addition, players of different heights may have their most comfortable height range for the physical crouch control. Perhaps the threshold thus needs to be personalized rather than hard-coded. In addition to the threshold problem, the crouch-based activation may also limit the player's view in the game.

To be a bit verbose on this topic, I think it is difficult to engineer a height- or crouch-based control technique that works like a physical input. From the hardware perspective, a button or a joystick is engineered and ergonomically tuned to have the force feedback, the dead zone, and the clear states of pressed and released. However, our head and body do not work like that, nor do the controls based on their movements. Therefore, I suggest that we be cautious and provide additional scaffolding when using the analog body positioning as state triggers to prevent them from being unstable and problematic.

In contrast, another sliding example using the affordance of a zipline simplifies the control. As a result, it achieves better efficiency than the error-prone physical crouching. Players can reach out and grab the zipline with either hand and hold the trigger button to slide along it (Figure 75).

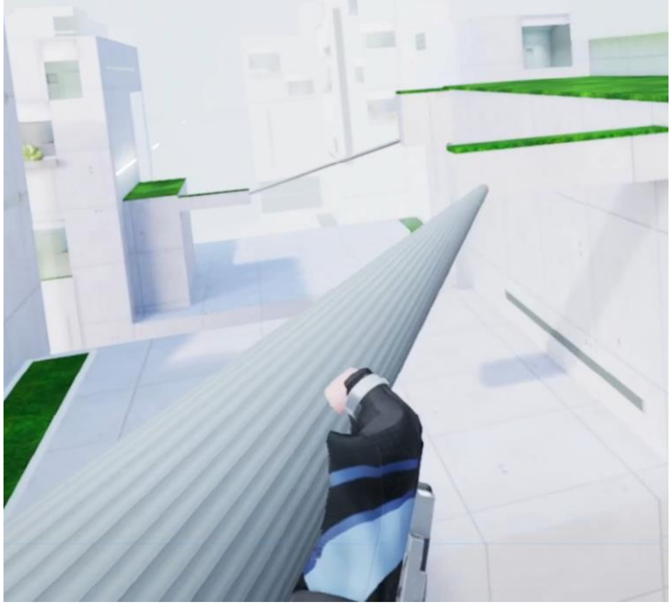


Figure 75. A zipline for grabbing and sliding in *STRIDE*

This simple example achieves both simulation and efficiency. Inspired by this special case of sliding, I think a similar design can be applied to sliding on the wall or any surface using simple button pressing for efficiency and affordance-based design for simulation. However, the limitation is still on the platform side that only affordance for the tracked hands and the head can be used in such design, while the slide action is more commonly mapped to the feet rather than the hands.

4.5.2. Open Arms in Mid-Air

The glide action can be mapped to opening arms in mid-air, such as the case in the well-received game *Population: ONE*, and it achieves a similar physical effect like sliding, but a more awesome effect – flying, as shown in Figure 76.

Walk off of the edge and hold your arms out in a T shape to fly

Figure 76. Glide interaction in *Population: ONE*. With arms open in a T shape, players can fly in mid-air

This interaction is easy to perform both in seated and standing modes as long as there is enough space for the player to open arms in a T shape. It is an accurate simulation of the skydiving action that is what the game starts with. *Population: ONE* is a VR battle royale game like PUBG and Fortnite, in which players start with skydiving from an airplane to land on an island to fight until they become the last person left. In the actual game, this interaction is also useful to fly between buildings and towers to collect resources and get in good positions. In addition, players can still look around and look down below while gliding.

We can see the other character in Figure 76 from the third-person perspective that their body is floating. However, I did not have any uncomfortable feelings or disconnection from the game experience when I did this in the first-person perspective because of my body not floating like that. I think this is an interesting literacy developed in VR gaming that I tend to only care about the matching of my hands and head’s movement, but for other parts of the body, as long as they have minimal movement during the interaction in the game, the mismatch in their position become trivial to the overall experience.

While this section is titled with slide and glide, I want to include the mid-air flip interaction in *Sairento VR* that I mentioned in the jump section for its using a similar technique of opening the arms in mid-air. In terms of the body orientation, this one’s mismatch with the real-world body is more dramatic as it can flip 180° once the player reaches the ceiling then attack enemies from above. As shown in Figure 77, instead of opening the arm in a T shape, the player reaches the arms above the head to perform the backflip action. While it is a good simulation of the backflip with the bonus of dodging bullets, it can cause a high level of disorientation[20].

Figure 77. Reach arms above the head to perform a backflip in *Sairento VR*, screenshot from: https://www.youtube.com/watch?v=Mt5YxdmoX40

---

[20] Even this player has a master-level of skills and performs the moves fluidly, there is an occasion when this player says in the video “that was a bit disorienting.”

4.5.3. Summary

While this section's examples are not strictly centered around a single semantic action, they share commonalities in terms of the physical concepts and interaction techniques.

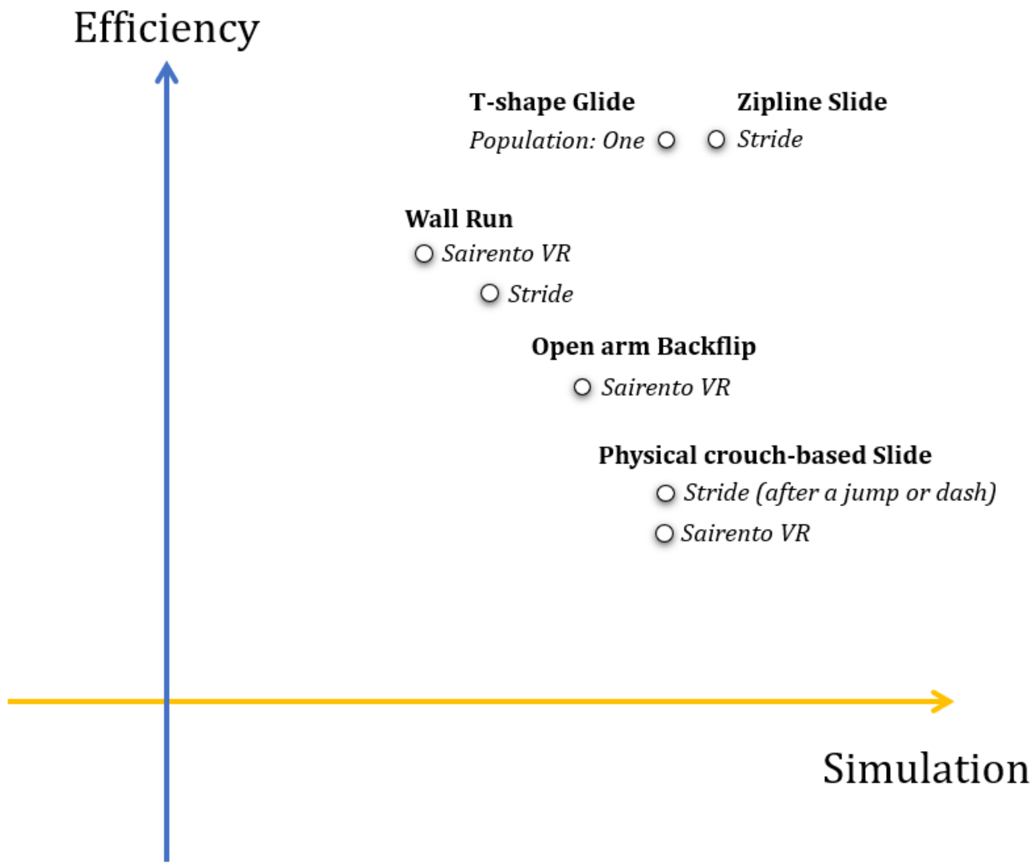


Figure 78. Summary of slide, glide, wall run, and flip examples

As shown in Figure 78, overall, these examples achieve a moderate or high level in both values. The zipline slide example and the glide example stand out in my theoretically optimal top right corner. Although the T-shape arm technique cannot fully simulate the floating body, it matches the typical skydiving posture and becomes a unique interaction on its own. Also, these two examples require minimal body movements, making them efficient to perform in the gameplay.

In the next section, we will have a focused discussion on the climb action, which is a commonly used locomotion technique in VR games.

4.6. Climb

Most VR climb interactions are enabled by grabbing and alternating hands, which already achieve at least a moderate level of simulation and efficiency. Still, the examples that we will see reveal value tensions in how they handle the details in the interaction design.

4.6.1. Examples

Boneworks: High physics simulation of the rigid body, low efficiency and visual simulation

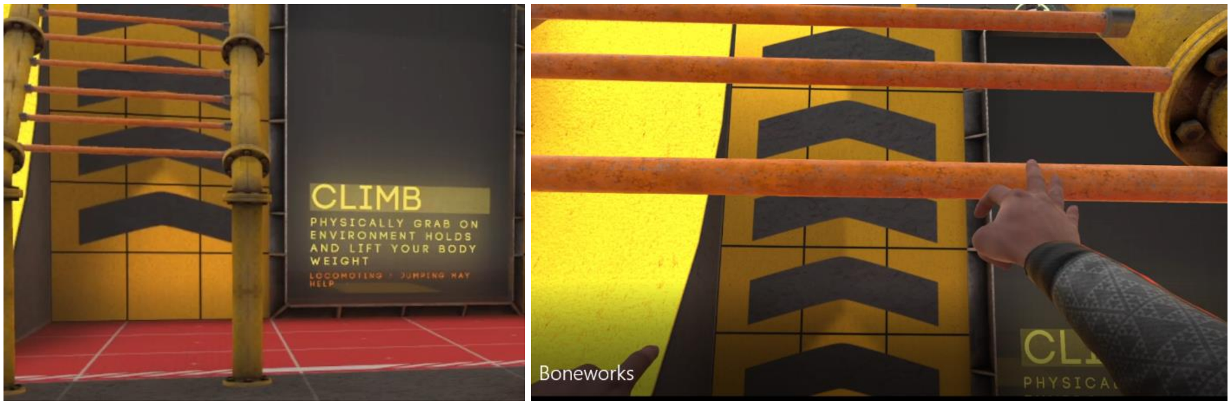


Figure 79. *Boneworks* climbing example

As shown in Figure 79, the first climbing scene in *Boneworks* is to climb over the bars on an exerciser. However, after I got over a couple of bars, the tilted angle and the rigid body prevented me from easily going up further. The game features a high physics simulation, but in this case, the bodyweight makes the climbing inefficient.

Another design detail is on the visual feedback of the hands when climbing a pipe, as shown in Figure 80. Sometimes the wrist angle can be distorted to show a poor visual simulation. This distortion can be due to the VR platform constraint that only tracks the hands and not the arms or wrists. Because this game wants to achieve a realistic body and physics, the visual design has to “guess” or model what the arm and wrist angles will be like, and as a result, they sometimes appeal unnatural.

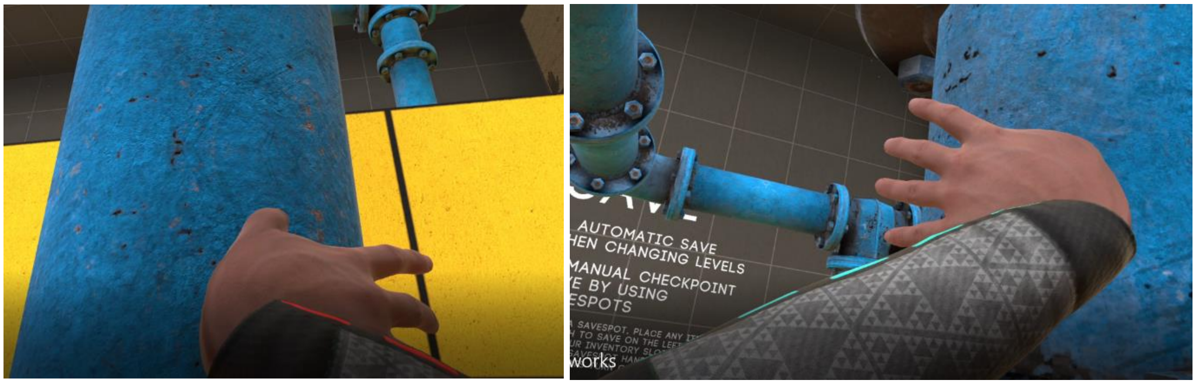


Figure 80. Climb up a tube in *Boneworks*, the hands appear distorted

Legendary Tales: non-body simulation and unrealistic positions of getting into a ladder, but high efficiency in climbing a robe

As a comparison example, in *Legendary Tales* (Urban Wolf Games Inc., 2021), the game does not simulate a full rigid body of the player but only the head, hands, and slots for storing weapons. Consequently, players can rotate around their grabbing hand to move into unrealistic positions when climbing the ladder, such as moving the head inside or passing through the ladder, as shown in Figure 81.

Figure 81. Unrealistic positions that players with an empty body can accidentally reach while climbing the ladder in *Legendary Tales*

On the other hand, the game's robe climbing reveals the efficiency of not simulating the body and weight. In Figure 82, we can see that players can climb the robe at any angle without the simulation of torsor and body. Thus, they do not need to worry about falling off a horizontal robe and grabbing it with their hands reaching upwards above their head.

Figure 82. Climbing on robes (vertical and horizontal) shows the benefit of not simulating a rigid body and weight. *Legendary Tales*

Population: ONE further optimizes climbing by adding visual highlights for the hands that have to reach inside the wall to start climbing.

Figure 83. An optimized design of climbing that requires the hand to reach inside the wall

With the design shown in Figure 83, the game can avoid the “distorted angle” problem in the *Boneworks* example by transforming the requirement of simulating the physical grabbing touchpoints and wrist angles on a round tube or any climbable surface into a simple implementation of reaching the hand into a flat surface. Although it loses the simulation of the hands firmly grabbing and sticking to the object, it makes climb action efficient.

In addition, the game still preserves the whole body and avoids the Legendary Tales' problem of the head moving inside the ladder. Because the red tube is only a visual indicator of climbing and there is no ladder or any gap, the solid wall surface can prevent the head from moving into it.

Figure 84. The pulling forward interaction at the end of the climbing to land on the ground, *Population: ONE*

Another efficiency-related design detail in *Population: ONE*'s climbing example is how players finish climbing by pulling themselves forward at the edge of the ground. Upon release of the pulling action, the player's body automatically snaps on the ground at the end of the climbing.

Figure 85. Climbing in *Half-Life: Alyx*

Half-Life: Alyx visualizes the hands only but still achieves stable body movement during climbing. It also has a simplified finishing animation without pulling forward when the hand reaches the ground (Figure 85). These details again reflect the priority of efficiency over simulation in this specific interaction. In my opinion, this is an optimized design because climbing is not a key game mechanic.

The Climb 2 (Crytek, 2021) is a game that does take climbing as the most important game mechanic, as its name suggests. Although it is not an FPS or RPG that I focus on in this study, it shows additional features of climbing that I feel are worth mentioning in this section.


(a) (b) (c)

Figure 86. Additional stamina mechanics and visuals in *The Climb 2*: (a) the balanced state with the trigger button pressed halfway; (b) losing stamina when the trigger button is fully pressed; (c) a shaking hand mechanic to recover stamina by pressing and holding the grip button. Screenshots from https://www.youtube.com/watch?v=sv25OnFJ2oQ

As shown in Figure 86, the game simulates stamina mechanics in its professional gameplay mode. It visualizes stamina on a ring-shaped bracer on each wrist and a grabbing force indicator as a blue disk that floats up and down depending on the degree of pressing the trigger button. The goal is to keep the blue disk within the range of the ring to keep a healthy stamina state by pressing the trigger button only halfway. If players keep the trigger all the way down, the disk will fall out of the range, and stamina will start to decrease. Fortunately, players can recover stamina by shaking the hand at any time, and there will be a freezing visual effect that simulates the cooling of a tired hand.

Together, these additional features enrich the gameplay interaction with climbing and achieve high simulation and efficiency values. The only control that was a bit counter-intuitive is pressing and holding the grip to shake the hand and recover stamina, while the shape of the virtual hand does not match the physical hand. I could rationalize it by assuming the freezing effect results from squeezing something equipped in the glove. However, only detecting the physical hand flipping or shaking can lead to false-positive results, so adding a button action can be efficient in this regard. Still, I felt it lacks a bit of simulation value with the visual mismatching.

In terms of the climbing mechanic, the game snaps the hand to align with the grabbable object when they are close enough. Technically, it is an implementation of setting the collider size of the grabbable object a bit bigger than it looks to ensure efficiency without losing much simulation value. Also, every time the snapping triggers, there is a distinct sound effect to ensure players have safely grabbed the object and are ready to move.

4.6.2. Summary

With the standard control techniques of grabbing and alternating hands, climb interaction can easily achieve high values in both simulation and efficiency, as long as it is free of visual, physics, and UX problems. The mapping of the examples in Figure 87 presents my evaluation of the importance of some design details that affect simulation and efficiency.

First of all, I find that not simulation the body visually is not a significant problem for efficiency and the overall immersive experience, but different ways of handling the body physics can yield mixed results in both values. Also, with the same simulation setting for the control techniques, efficiency can still vary with the climbing object (see the ladder and robe examples in *Legendary Tales*). On this issue, *Population: ONE* offers a simplified and uniform design for any climbable surface at a minor cost of simulation. Overall, I find the simulation value is primarily affected by the visual matching, and the efficiency value is primarily based on the overall speed and robustness affected by the physics and mechanics.

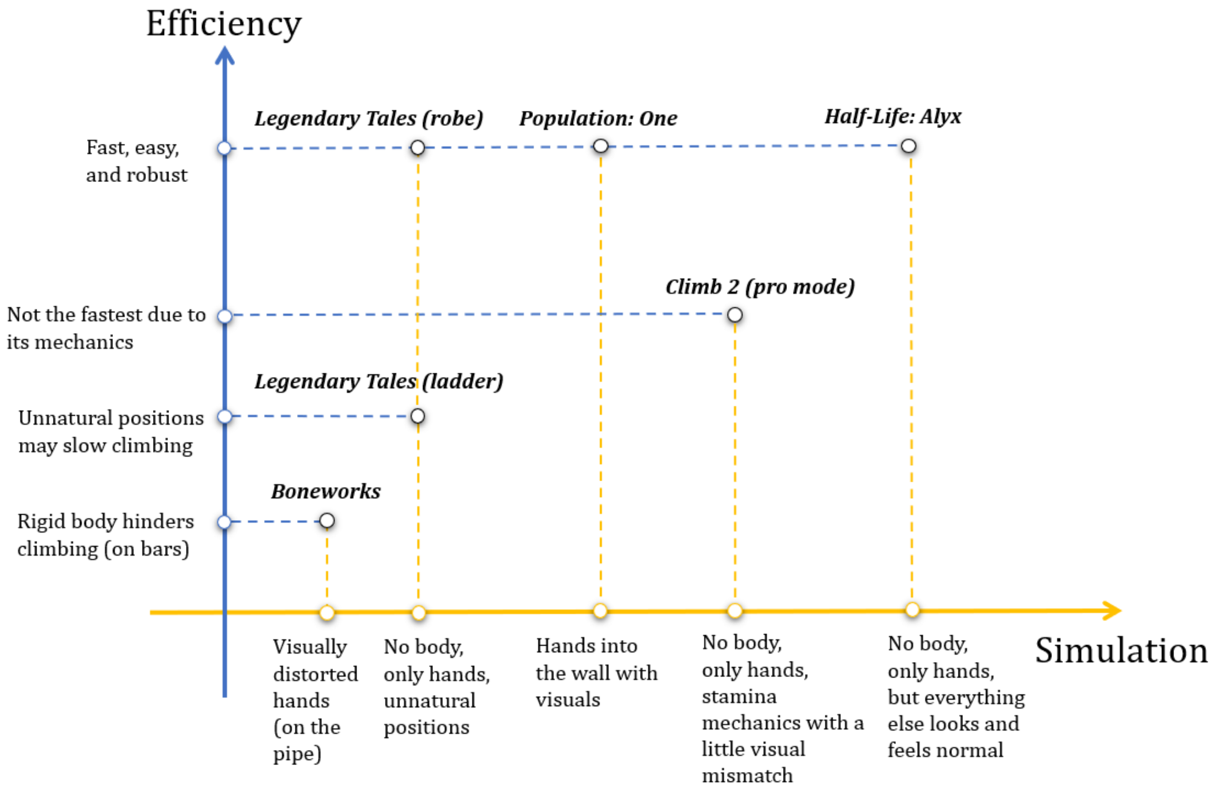


Figure 87. Summary of climb interaction examples

4.7. Special Locomotion Design Cases

So far, we have reviewed some most commonly used semantic locomotion actions and their interaction design that reveals various levels of simulation and efficiency. This section picks up some special actions that are less common and more difficult to fit within the abovementioned categories. Still, I think they show potentials for future and situational uses considering they achieve high simulation and efficiency values.

Also, we will take a step back from the detailed design forms and implementations to look at how specially designed control schemes and options incorporate multiple locomotion interactions and reconcile any of their value tensions.

4.7.1. Zero-Gravity, Swim, Hook

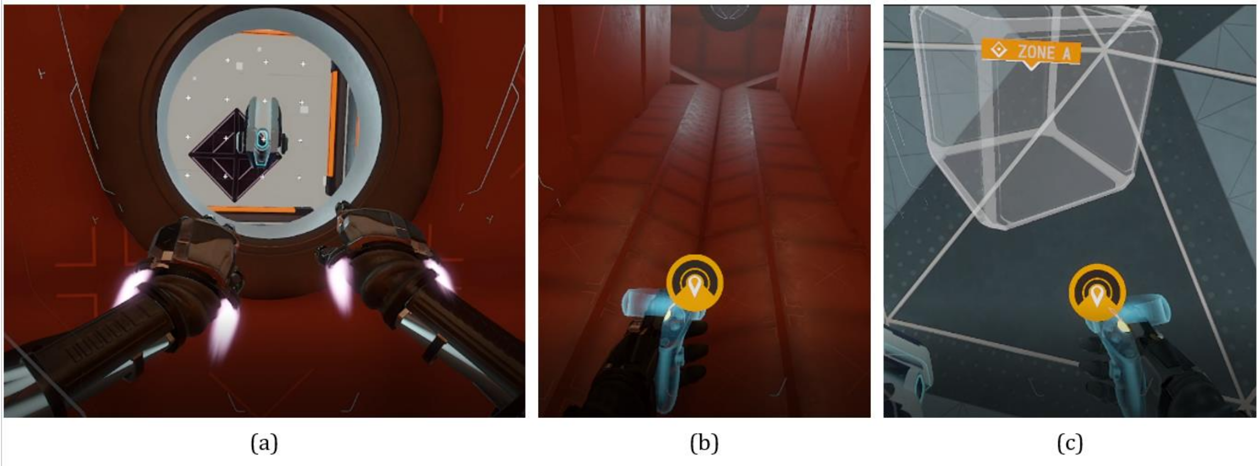
Zero-gravity (zero-g) movements, swimming, and hooking have use cases for locomotions that are not on a ground surface. Zero-g movements are featured in the *Echo*

VR (Ready At Dawn, 2017) game series as their fundamental locomotion mechanics. For example, players can grab and push any surface in the zero-g environment with their hands and move themselves using the pulling or pushing forces that follow the laws of physics. As shown in Figure 88, the hand and the arm in *Echo VR* show their contours behind objects to improve visual efficiency for this fundamental move interaction. This can also translate to climb action and makes it feel easy in the zero-g environment.

Figure 88. Grab to move forward or climb in a zero-g environment, *Echo VR*

(a) (b) (c)

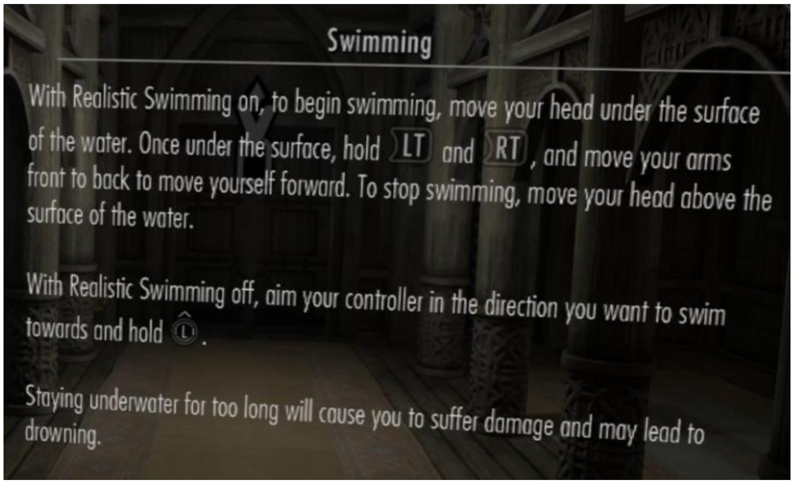
The game gives players additional boosting and braking controls to gain immediate momentum while floating. As shown in Figure 89, on each hand, a thruster can be activated by holding the B or Y button to gain momentum. One benefit of this design is that it avoids any guessing of whether this movement is based on the head or the hand direction – since it is visually equipped on the hand, it works as it looks like, moving the player in the direction where that hand is pointing towards. Besides, players can click the left joystick to get an even stronger boost in the head direction and click the right joystick to break. The thruster and the boost option present a unique solution to allow hand- and head-based movements to coexist, which often players have to choose one or another in many other VR games.



(a) (b) (c)

Figure 89. Additional movement techniques in Echo VR's zero-g environment: (a) Hand-based thruster; (b) boost in the head direction by clicking the left joystick; (c) use breaks to stop in a zone.

A drawback within efficiency is that the game does not have any movement controls mapped to the left joystick since there is no "direct movement" in the game's accurate simulation of the zero-g environment. This control scheme can be a little unusual and inefficient since clicking the joystick button is more difficult to perform than pushing it. Moreover, it can hinder the player's attempt to press the Y button simultaneously with the thumb. Still, the zero-g movements in *Echo VR* have presented an overall optimal solution that achieves high values of simulation and efficiency. It works well to support the game series' well-received single-player adventure gameplay and the multi-player combat arena games.



Swimming

With Realistic Swimming on, to begin swimming, move your head under the surface of the water. Once under the surface, hold LT and RT, and move your arms front to back to move yourself forward. To stop swimming, move your head above the surface of the water.

With Realistic Swimming off, aim your controller in the direction you want to swim towards and hold L.

Staying underwater for too long will cause you to suffer damage and may lead to drowning.

Figure 90. Swimming controls and the realistic option in *Skyrim VR*

Like zero-g interactions, the realistic swimming option (Figure 90) in *Skyrim VR* is another interaction example in a special environment – underwater. Revisiting the realistic swim option that I mentioned in the introduction hook of this dissertation, I find this control has a high simulation value because of interaction fidelity, although it cannot track the feet movement to mimic the motion of all the body segments in real-world swimming. Like the experience with the T-shaped gliding example in *Population: ONE* and the movement interactions in zero-g, the floating body that remains relatively motionless does not affect my immersive experience or judgement of stimulation so much as the hand movements. With the hand movements, I can feel the rhythm of moving forward, simulating the real swimming.

Figure 91. A player uses the realistic swim option and a special treadmill VR input to enable the full-body swimming simulation. Screenshot from https://www.youtube.com/watch?v=dsbGKiYkV7Y

The example shown in Figure 91 uses a more expensive treadmill VR input to support full-body swimming simulation. Although the game does not track the feet or body orientation, the higher embodiment can make exploring the underwater world more enjoyable.

However, the game currently does not encourage or reward the player for doing so. If the realistic swimming speed was set much higher, or there were more fun things to do underwater besides catching a random fish, I would see more efficiency value for this option. Still, it remains an opportunity for future game design to extend this option. As an audience user points out in the video, there is still room to tweak the swimming interaction for realism by adding the inertial motion of floating forward before the stopping control.

The last special case of locomotion interaction that I want to include in this chapter is the hook interaction. Thanks to the spiderman movies, this interaction is well received among the VR community.

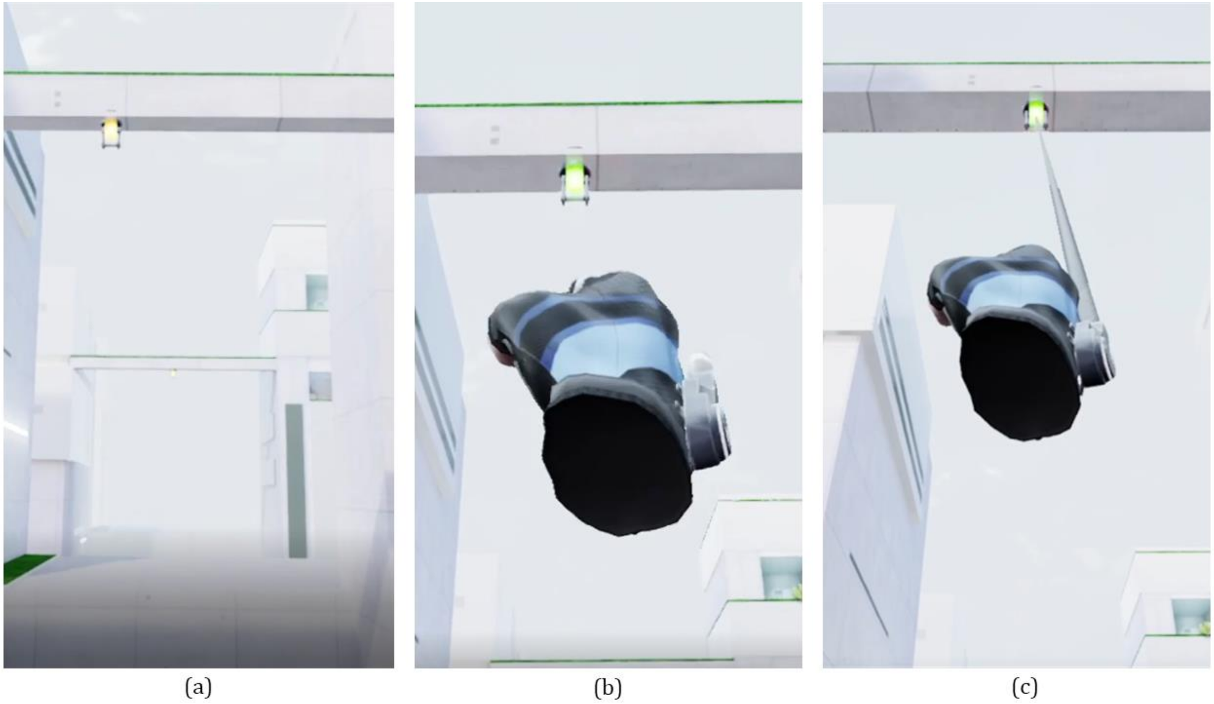

(a) (b) (c)

Figure 92. Hook interactions in *STRIDE* with a color-coded node to link with: (a) the node starts in the yellow color; (b) the node turns green when the player reaches out a hand to aim at it to confirm it is ready to link; (c) the player shoots the hook to link with the node by pressing a button

The *STRIDE* example of hooking uses a node with color feedback to link with the player's hook and signify when players have aimed at it and are ready to link (Figure 92). Unlike PC platforms with accurate pointing or aiming, in this case, aiming accurately at a small node while the player is in mid-air is not that easy. For this reason, the game system implements an aiming assist mechanic in addition to the visual feedback to ensure efficiency.

The entire gameplay of *SwarmVR* (Greensky Games, Inc., 2020) is based on hook locomotions. As shown in Figure 93, the spheral arena is designed with many platforms that afford a high chance of hitting one with the hooks that players can shoot in both hands to get high in the air and shoot enemy swarms. After linking with a platform, players can pull the controller back to increase the tension on the hook for going higher. In later gameplay, the hook is upgraded with the new zipline function, allowing players to pull themselves towards the platforms or enemies (Figure 94). This option further increases the efficiency of moving in this game.

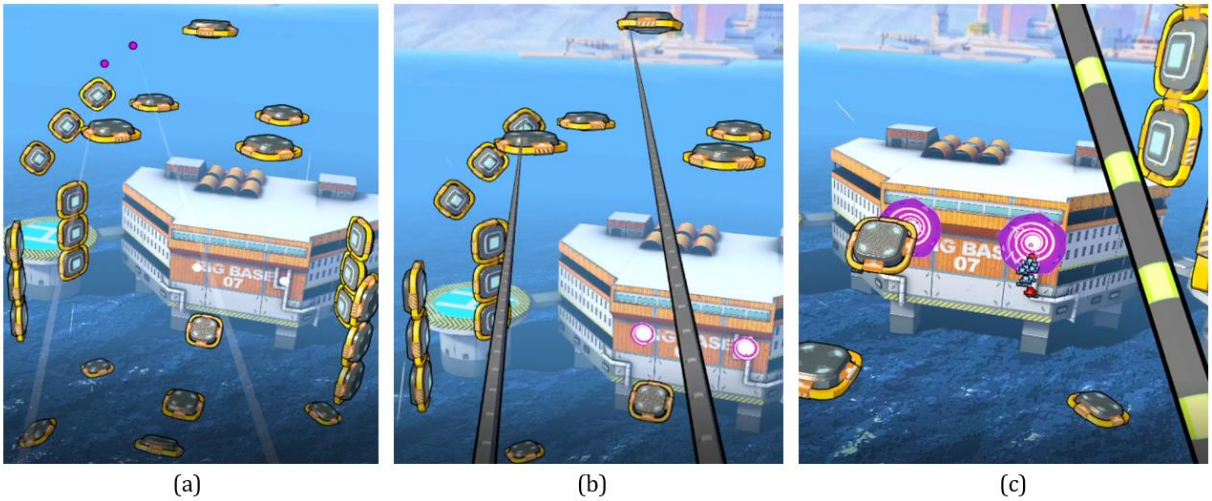


Figure 93. Hook interactions in *SwarmVR*; (a) Point at a platform; (b) Shoot hooks; (c) hooks are colored to indicate the tension

Figure 94. Upgraded hook interaction with the zipline function to quickly pull themselves towards the linked platform. Screenshots from *SwarmVR*

In summary, the zero-g moving techniques, realistic swimming, and hook interactions present special design cases that achieve high values in simulation and efficiency by leveraging the unique affordances and physics in the game environment. They enrich the design space of non-ground locomotions, and future VR game design practices can further incorporate them into the locomotion systems for FPS and RPG products.

4.7.2. Reuse Control Schemes

When reviewing locomotion interaction examples, I find a special case in which part of the game's locomotion control schemes changes depending on the basic movement options.

As we discussed, *Half-Life: Alyx* reused the same control (right joystick up direction) for jump and mantle in different situations. It also handles the crouch and stand actions with the left joystick click and hold depending on which option the player chooses. The same design philosophy extends to how the game uses the available controller inputs under different movement modes: teleportation and continuous movement.

Specifically, when continuous movement is chosen as the primary movement technique, the joystick down direction is used for a quick blinking movement like teleportation. However, when teleportation is set as the primary movement technique, the joystick down direction changes to a quick dodge back action.

This special design of control schemes, again, increases the efficiency of performing these locomotion actions. VR games typically force players to choose between teleportation and continuous movement; however, these two options can be useful in specific situations. With this design, players favoring continuous movement do not have to open the menu and change options to use teleportation. Players who prefer teleportation as the primary technique gain the agency to push the right joystick down to dodge in the back direction. This action otherwise will take extra steps and time with turning around then teleporting. Considering the gameplay where the player often encounters biochemical creatures jumping towards their face, the dodge action is efficient also in the game context. The direction of dodge back also matches the joystick down direction, adding a bit of value to the interaction fidelity and intuition. In addition, the remapping of an overall less commonly used controller input to enhance both movement modes saves the already limited control inputs. It is more efficient than mapping these actions in both modes.

5. Object Interactions

After players are familiar with the locomotion interactions, they want to interact with the objects in the VR environment. This chapter serves as a transition by reviewing common object interactions to support more complex combat interactions, which will be discussed in the next chapter. Specifically, these object interactions are centered around the *player body*, which can become a design space for embodied inventories and equipment slots after players have grabbed objects from the environment.

5.1. Grab

Grabbing is perhaps the most commonly used object interaction in VR. Although the grab action can translate to different meanings, I specifically focus on the interaction design of grabbing game objects such as weapons and ammo.

The examples are divided into Direct Grab and Remote Grab based on their difference in interaction mechanics. Direct Grab uses the direct collision of the hand to grab an object, while the Remote Grab can pull objects in the distance via a force without the direct touch.

5.1.1. Direct Grab

As for direct grab interactions, I will focus on the following design aspects and issues: the interaction states of selecting and grabbing and visuals on the objects to indicate the states, the cases of using both hands to grab an object, and the issues of dropping the weapon from the hand, which will have a negative chain effect on the values in other game actions as well.

5.1.1.1. Interaction states and visuals

The objects in the game scene usually have special visuals to draw the player's attention and provide feedback for their selection states. In the *The Walk Dead Onslaught* (Survios, 2021) example shown in Figure 95, the ammunition and supplies available in the environment have a visually contrasting highlight for players to not miss them in the game contexts of an overall dark environment and theme of zombies and surviving. When the player is close to these objects, each object's type and the amount is presented by pins and icons. When the player hovers their hand close to an object to select it before grabbing, the object shows a ring to indicate its state of being selected.

Figure 95. Grabbable objects that signify themselves and show informative icons when the player is close in the game *The Walk Dead Onslaught*

These visuals are non-diegetic and, strictly speaking, not a realistic reflection of how grabbing objects in the real world looks like. However, they become a standard design pattern in many VR FPS and RPG to improve the efficiency of grabbing interactions. In the game *Population: ONE*, the ammo grabbing interaction features a similar visual design (Figure 96). Each grabbable object has a circle on top of it. When an object is selected, it shows its name and icons of two hands, and the circle turns green and bigger, signifying the action of grabbing the object.

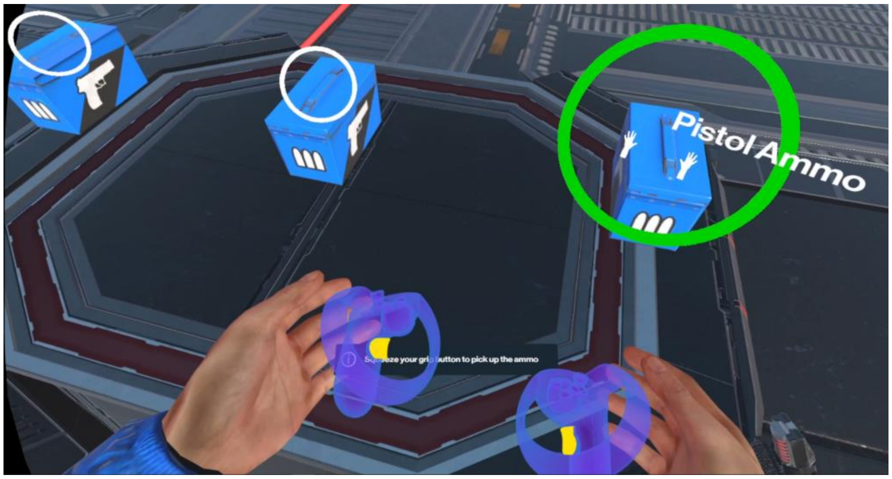
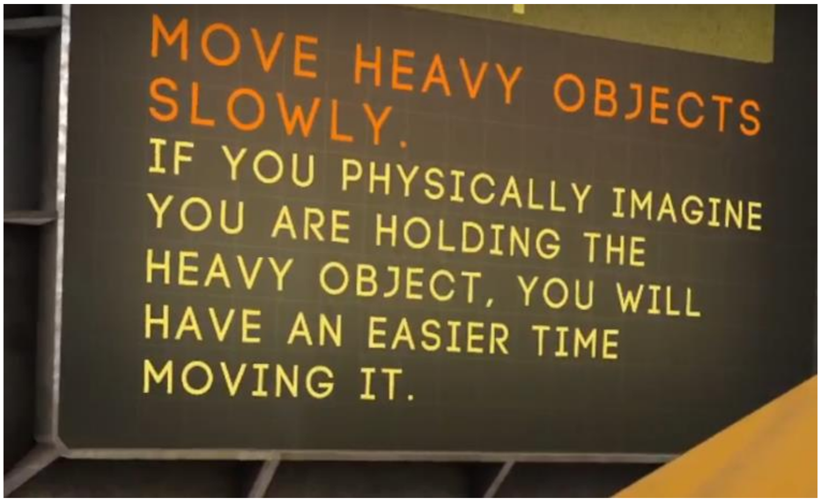


Figure 96. Grabbing ammo in *Population: ONE*. The circle on the ammo object turns green and bigger once it is selected and ready for grabbing.

In *Skyrim VR*, when grabbing a weapon, its relevant game information, such as damage, weight, and selling value, are presented to players in addition to its name and the control (Take) action (Figure 97). Compared with the circle indicator, I find this visual and textual information is more efficient in helping players decide whether to grab the item or ignore it. The circle indicators in the above examples may be helpful for small objects, but it does not give textual information. On the other hand, the bow in this example is large enough to be visible to players while having a realistic look that matches the game world.

Figure 97. When grabbing a bow in *Skyrim VR*, information about its values in the game context is shown to the player

5.1.1.2. Both hands



MOVE HEAVY OBJECTS SLOWLY.
IF YOU PHYSICALLY IMAGINE YOU ARE HOLDING THE HEAVY OBJECT, YOU WILL HAVE AN EASIER TIME MOVING IT.

Figure 98. Instructions for moving heavy objects with two hands in *Boneworks*

Heavy and big objects, such as a block and a puzzle object, reflect the simulation of using both hands to grab them and the inefficiency of having a slow movement speed.

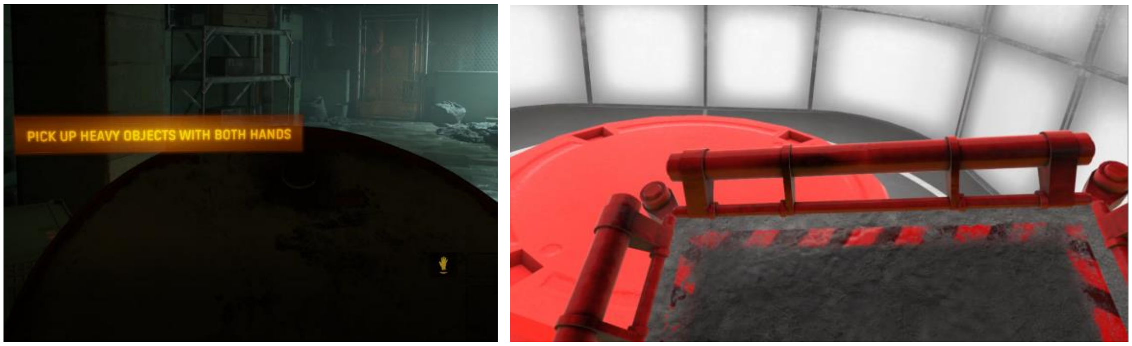


Figure 99. Grabbing heavy objects with both hands. Left: a barrier block in *Half-Life: Alyx*. Right: a puzzle object in *Boneworks*, when grabbed, the player moves slowly.

This is a case where I think efficiency is traded for simulation. Although it is slow, it feels realistic and reinforces the narratives of making a way out in the game environment and solving the weight puzzle, as demonstrated in the two images in Figure 99, respectively.

5.1.1.3. Drop Weapon?

The biggest drawback of direct grabbing is perhaps when the grabbed object is dropped on the ground that is out of the player's reach to grab it again. Although dropping is the reverse interaction of grabbing in semantics, the meaningful use cases of dropping are much more limited than grabbing. In some cases where I dropped an item, they were totally by accident but could amplify some negative consequences.

For example, some games require the player to keep holding the weapon at hand during combat; otherwise, it will drop on the ground. The player has to either keep holding the grip button or avoid pressing it to untoggled the grabbing. Either option is prone to drop the weapon by accident. In many games, the original place where the item is grabbed is designed to be close to players, such as on a table. However, the floor that the item is dropped onto is usually out of the player's reach without physically crouching. Worse still, it may be hard to estimate how far the item is away from the hand when looking down on the floor.

Figure 100. Physically crouching to grab an accidentally dropped weapon is inefficient in the gameplay context. Left: grabbing a sword on the ground that is out of the player's reach, *Blade and Sorcery*. Right: Having to grab the dropped pistol in the FPS game *Pavlov VR* while taking damage and bleeding

In cases of not having a crouch functionality, performing a physical crouch to grab the accidentally dropped weapon is inefficient (Figure 100). Worse still, in the context of taking damage under enemy fire and seeing myself bleeding, I hated so much that I still had to look down on the floor for my pistol and failed my attempts to grab it because I had no idea how far it was from my hand (Figure 100, Right).

My experiences with such cases reveal the conflicts between simulating the realistic drop mechanics of the weapon and the efficiency of using it in combat. The platform aspect of either holding the grip button forever or never pressing it to avoid accidentally dropping the weapon adds to the tension, and so do the game contexts of being under a time urgency and enemy attacks. From a general UX point of view, the strict simulation of the drop mechanics in the example violates the rules of “prevent errors” and “permit easy reversal of actions” (Shneiderman et al., 2016). Therefore, I advocate for efficiency in this case to have an easy mechanism of holding the weapon over the realism simulation of dropping.

There are a few ways to reconcile this tension. For example, in the game sword-fighting game *Until You Fall*, dropping the weapon in the game is not an option. Instead, players can toggle the grip button to vanish and summon the weapon again magically (Figure 101). The only case where players can change a weapon at hand is in the respawn room, where they can purchase and upgrade weapons. However, this game is a special case, and, strictly speaking, its magical effect does not count for the grabbing comparison.

Figure 101. Weapon vanishing (left) and summoning (right) by toggling the grip button, no need to hold the grip button and no dropping. *Until You Fall*

Other ways to avoid the dropping problem includes remote grabbing, having the weapon “drop” automatically into a holster or a body slot, and switching weapons on an inventory interface. We will discuss them in the rest of this chapter.

5.1.2. Remote Grab

Remote grab, or force grab, typically uses a hyperreality pulling force to grab a remote object close to the hand. However, the detailed interaction design still varies from case to case.

To remote grab an object in *Boneworks*, players first reach their hands to target the object, then press the trigger and grip buttons together to pull the object. A small visual reticle (the yellow dot as if it was the toy's nose in Figure 102, bottom left) appears as the visual feedback when the target is selected. There is also a sound effect of airflow as the audio feedback.

Figure 102. Force (remote) grabbing controls and an example of grabbing a toy in *Boneworks*

The hand's reaching out action seems redundant and inefficient, but it triggers the state of remote grabbing instead of direct grabbing. Remote grabbing is not something we can do in the real world unless you are the spiderman or have magic. So the simulation is about making the unreal feel real and simulating the concept of the action in science fiction. In this sense, I think this game action is of moderate to high simulation because it involves target selection and grabbing from the remote. The minimum visual reticle preserves the overall realistic immersion of the virtual world without being too prominent during the interaction. However, sometimes not seeing the reticle negatively affected my experience because the action only took effect within a certain distance, and the reticle helped confirm the range. Also, as shown in Figure 103, the reticle visual seems to vary from object to object. On a white cup, it even appears to be invisible due to the color. As a result, I find that on occasions, I had to pause a moment to look for the reticle; otherwise, I would risk failing to grab it because I was too far away from it.

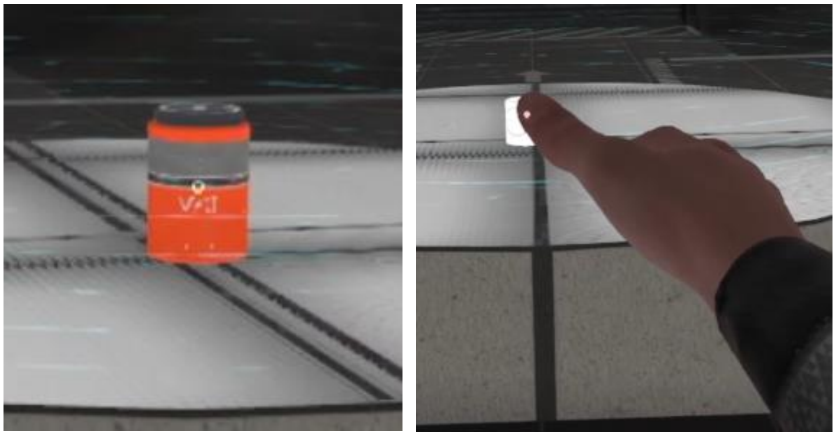
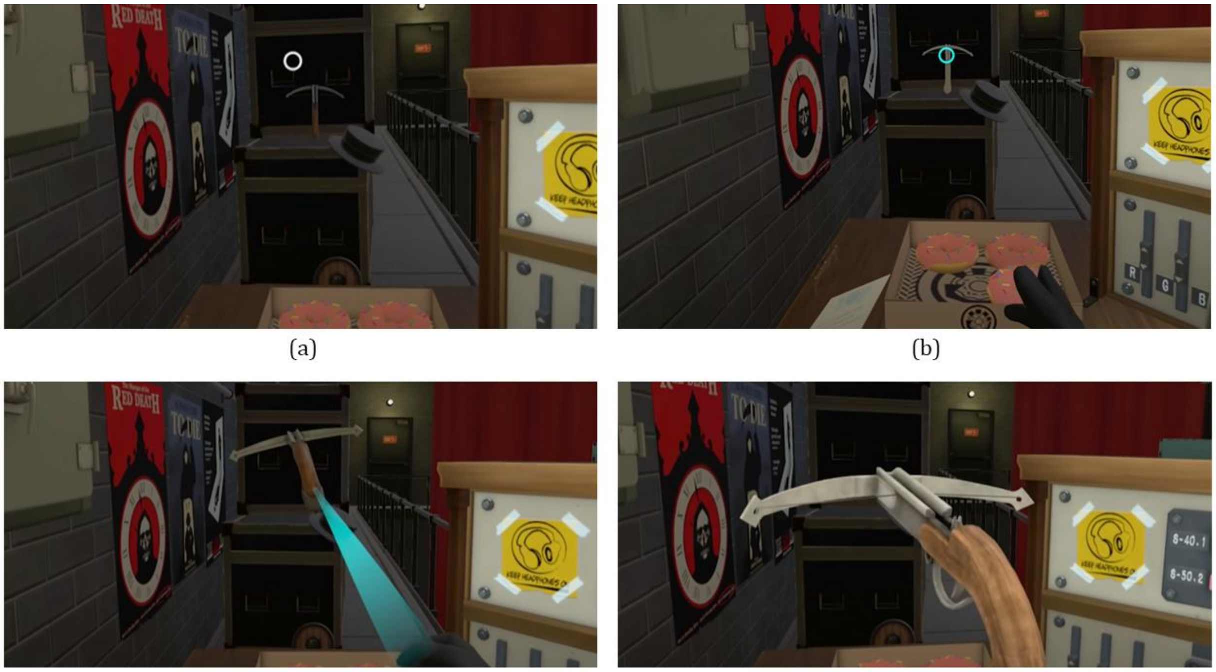


Figure 103. The reticle for remote grabbing varies from object to object and can appear nearly invisible on some objects.

The following case in the game *I Expect You To Die 2* (Schell Games, 2021) serves as a comparison example. It is not a typical FPS or RPG combat game but is more themed with adventure, storytelling, and puzzles. That is to say, a lot of times in the gameplay, the hand is empty and not bound to an item or a weapon.



(a) (b)

(c) (d)

Figure 104. Remote grabbing phrases and visuals in *I Expect You To Die 2*: (a) pre-selection; (b) selection; (c) grabbing; (d) finish.

As shown in Figure 104, the white reticle in the pre-selection phase turns blue in the selection phase. It appears large enough to be visible and small enough to target an object, in this case, a crossbow, in the distance exclusively. Once the object is picked up in the grabbing phase, the player can pull it by holding a controller input. A line connecting the object with the hand signifies moving it along the line. When the crossbow is drawn to the hand, the hand disappears and becomes the crossbow. All of these design features reflect the efficiency value of object manipulation.

Figure 105. Remote grabbing a shield with no visual indicators in a sword-fighting and magic-themed game *Legendary Tales*

In *Legendary Tales*, the remote grabbing interaction is designed to be like a magical yet natural effect without any added visuals (Figure 105). It fits the artist style of an accident and dark environment and the magical theme. If the player grabs an object on the ground, it will be efficient enough even without any visuals as long as the player can roughly point to the object. Still, the fact that the object can be dropped on the ground by accident is inherently inefficient as far as I consider. The game implements the dropping action to allow players to pick up an alternative gear.

In *The Walk Dead Onslaught*, the connecting line looks like an airflow, with a light effect on the object (Figure 106). It sort of strikes a middle ground between simulation and efficiency and applies the "look and feel" design strategy (Bizzocchi et al., 2011) to fit the narrative of remote grabbing.

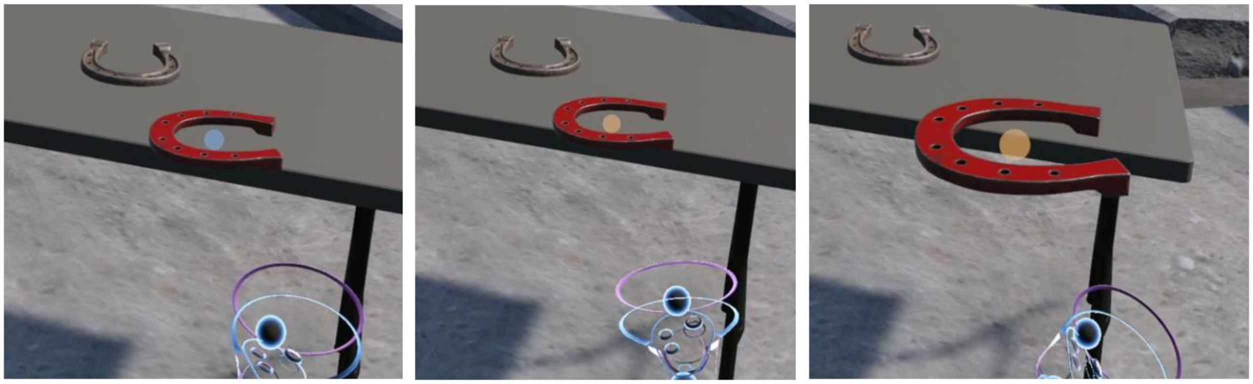
Figure 106. Remote grabbing and visuals in *The Walk Dead Onslaught*

The rest examples show alternative interaction techniques of remote grabbing. Figure 107 shows the novel hand flicking technique in *Half-Life: Alyx*.

(a) (b) (c)

Figure 107. *Half-Life: Alyx*'s hand flicking technique for remote grabbing: (a) point and press trigger to lock target; (b) Once tethered, press the grip and trigger to make a fist and flick hand back to pull object; (c) catch object in mid-air.

This example trades efficiency for extra simulation value by adding the step of flicking the hand back after the player has made a grabbing fist to simulate the instantaneous pulling effect. Although the learning curve is long (the game gives detailed tutorials and scripts to teach players) and is more prone to failure than the previous techniques, it adds a lot of dynamic feeling. It further allows the control of the direction in which the object is pulled to. The line only shows the end part connecting the object, and the bending angle indicates the direction in which the object will be drawn. Mastering this action actually made me feel "cool" in the gameplay.



(a) (b) (c)

Figure 108. "*Half-Life: Alyx* style" grabbing in *Hot Dogs, Horseshoes & Hand Grenades*

In *Hot Dogs, Horseshoes & Hand Grenades (H3)*, the developer literally uses "Half-Life: Alyx style" as the name of an option for its grabbing interaction. What the game implements differently in the design details, as shown in Figure 108, is getting rid of the line and instead only using a colored dot to indicate the selection state. Without the tether effect that enables the direction control, this technique simplifies the object's flying direction to be straightly going to the hand. This adds efficiency by removing the need to catch the object in mid-air. However, I feel it loses simulation value as the tether seems to be the fundamental supporting metaphor that this whole interaction technique is built upon.

5.1.3. Summary

With the number of examples and details covered in this section, it will be messy to lay out all the design aspects projected onto the simulation and efficiency continuums for comparison. So, for this summary, I choose to map the examples in groups to show both inter-group and intra-group comparisons (Figure 109).

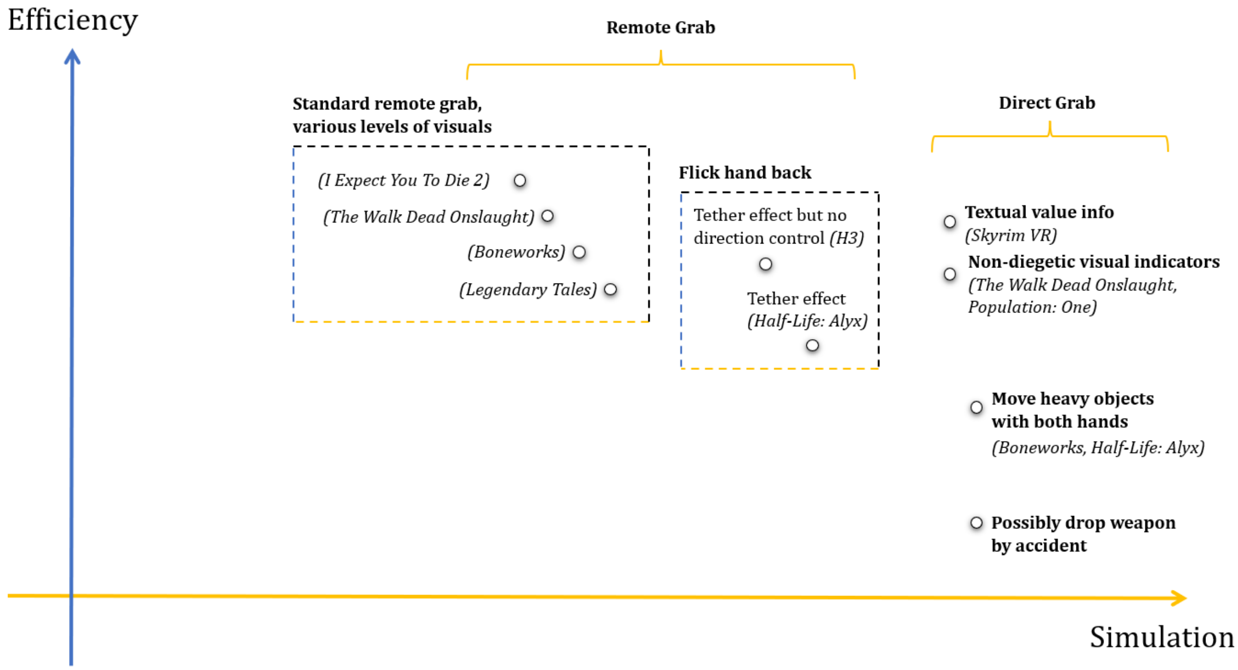


Figure 109. Summary and comparison of grab interaction examples on the simulation and efficiency framework, featuring overall inter-group comparisons and detailed inner-group comparisons.

First of all, I acknowledge that I could have made respective diagrams for the groups. However, since these examples all share the semantic meaning of "grabbing" or gaining objects from the VR environment, I might as well try my best to put them together and discuss the design space as a whole. I use colored dash lines to present two groups of examples sharing the same techniques while reflecting their simulation and values within the group.

At the high level, I contend that the commonly implemented remote grab interaction is a way to reconcile the inefficiency within the inherent drawbacks of the direct grab interaction, such as possibly dropping an item by accident to a position out of reach and having to move close to an object to grab it. Unlike how point-and-click interaction can work efficiently for both close and remote objects on screen-based platforms, VR promotes the embodied and physically simulated direct grab interaction for getting objects in close range, but it needs additional design of interaction techniques to scale to the remote range. Considering the remote grab interaction for easily obtaining objects, I map it higher in the efficiency continuum than the direct interaction.

Within the remote grabbing examples, I give the "flick hand back" technique a higher value in simulation than the standard approach because I think it simulates the physics of a string or a tether connecting the hand and the object. The standard approach does not simulate any physical medium between the hand and the object, making it like the interaction is through a pure magic force. However, because the "flick hand back" control requires more steps of specific hand movements than the standard approach, I rank its efficiency value lower to reflect its possibly longer learning curve. Furthermore, the *H3* implementation trades the simulation of the extra direction control in *Half-Life: Alyx* for the efficiency value of having the object always fly to the hand, just like the standard remote grabbing.

Finally, within the standard remote and direct grab interactions, non-diegetic visual indicators and feedback become the difference that amplifies the value tensions. The two-hand grabbing is a special case where I feel the simulation wins over the efficiency. Being slow in moving heavy objects feels realistic and reinforces the sense of achievement and narrative in the gameplay.

5.2. Body Slots: Store and Take

Figure 110. Showcases of commonly used body slots. Left: in *Blade and Sorcery,* the holster in the back is for storing a long sword. Right: various body slots in *Boneworks*

Body slots are a unique approach in VR that uses different parts of the body as containers for players to store and equip various items quickly. Figure 110 visualizes some commonly used body slots in FPS and RPG games. Depending on the game design, slots can be set at the head, the back, the shoulders, the chest, the hand, the wrist, each side and the middle of the waist, the hip, etc. However, the number of body slots and their effective areas (colliders) for interaction can still reveal the value tensions in various game contexts.

5.2.1. Examples

A common issue that I have identified in many attempts of storing an item into a slot is that players can miss the effective 3D collider of the slot and drop the item on the ground. This happens more often for side slots near the waist and the ones on the backside of the body.

Equipment Slot

You can equip 2 weapons on your hips and 2 on your back, for a total of 4.

Figure 111. Body (Equipment) slots in *SWORDS of GARGANTUA* (Thirdverse, 2019)

In my experience with the equipment slots in *SWORDS of GARGANTUA* (Figure 111), I had to spend a long time learning the size and the positions of the “hips” slots. Also, although they were called hips, I find that they were more precisely located on the sides of my waist. The game only provides sound feedback to indicate whether the weapon is stored or dropped on the ground when releasing over the hip position; however, the sound cues cannot prevent failure. The position mismatches reveal the VR platform’s limitation that there’s no perfect estimation of where they are exactly located. The VR platform can only track the exact positions of the head and hands but have no idea of the player’s *body shape*. As a result, some body slots can lead to both poor simulation and low efficiency.

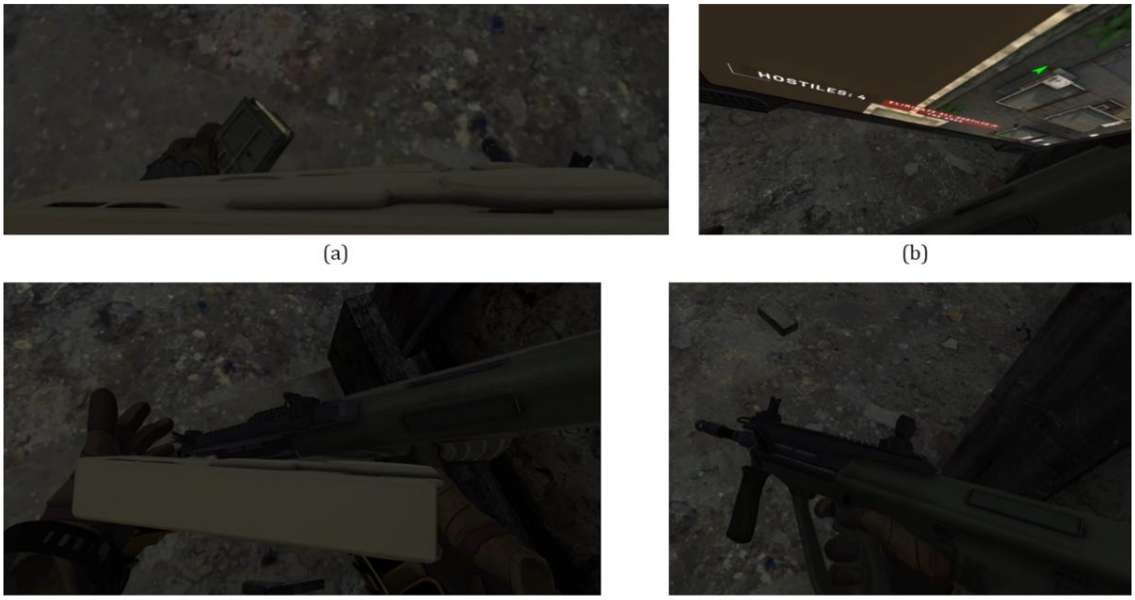
There are ways to fix this issue as well as ways to cause new problems. Let’s start with the worse cases first.

5.2.1.1. Amplifying the value tensions

In my experience in *Hellsplit: Arena* (Deep Type Games, 2019), I found this example where a heavy plate chest gear largely blocked my sight of the waist slot where I wanted to grab a potion. I can see the good thing of having the chest visual is to have players feel they are geared and in character, but this simulation lowers the efficiency of taking a useful item from the waist slot.

Figure 112. A plate chest gear largely blocks the sight of the waist slot. Screenshot from *Hellsplit: Arena*.

A similar case happened in the FPS game *Onward* with the chest container partially blocking my sight for confirming that I had effectively grabbed the item that I wanted from the side slots on the waist position (Figure 113). The large size of the rifle should have made it easier to store it into the side slot and take it from there; however, I find that the game only allows grabbing its handle, making it a more difficult task with the chest piece present.



(a) (b)

(c) (d)

Figure 113. Body slot interaction examples in *Onward*: (a) the chest container partially blocks the sight for confirming grabbing the ammo; (b) Unintentionally grabbing a tablet from somewhere on the chest; (c) Storing the rifle into the slot felt difficult because of the chest visual; (d) Taking the rifle was also difficult due to the strict condition that only the handle could be grabbed.

Legendary Tales has the waist side slots be more towards the front but leads to a new conflict in the context of grabbing an item in the game scene (Figure 114). This conflict is amplified when the hand, the slot, and the item are on the same line. The result was I could not grab the item on the ground because I would hit the collider of the body slot first.

Figure 114. The two side slots can conflict with grabbing an item in the game scene, especially when the hand, the slot, and the item (a bow in this example) are on the same line, screenshot from *Legendary Tales*.

In *I Expect You To Die 2*, players can wear a hat or a pair of glasses for fun and purely cosmetic uses (Figure 115). Although these props can transform into useful game mechanics in other contexts, they still show the potential disadvantages of partially losing the limited sight of the game scene. The frame of the glasses also does not accurately simulate how one would normally wear it without seeing it in the real world.

Figure 115. Wearing a hat or a pair of glasses reduces the sight of players. Screenshots from *I Expect You To Die 2*.

5.2.1.2. Reconciling the value tensions

One way to improve the efficiency of body slot interactions is to implement strong haptics feedback to inform players that the effective touch with the slot is confirmed. The haptics design, in this case, can compensate for the limitations of the visual and audio feedback and make the interaction more realistic. However, haptics on the controllers can still be hardly perceivable during fast motion, and there are other ways to improve body slot interactions.

A simple solution is to reduce the number of body slots so each slot can be larger and more separated from each other. This is like how Fitts' law[21] would apply to body slots. For example, the gameplay *STRIDE* only needs a body slot at each side to store a pistol (Figure 116). It seemed that the effective collider covers a large area of the torso, making it

---

[21] In HCI, Fitts' law is a classic theory on the speed-accuracy tradeoff within pointer-based interface design. It states that the time of accurately pointing to an 2D object is disproportional to its size and proportional to its distance from the user (Fitts, 1954).

easy and fast for one hand to move to any point within the effective collider at the opposite side to trigger the grabbing interaction. Although this design does not simulate an “accurate position and size” of a holster, I still feel the metaphor of the holster always being at my side and enjoyed the efficiency of getting the pistol.

Figure 116. In *STRIDE*, with only two slots at each side of the body, it is efficient for a hand to grab the pistol on the opposite side.

In addition, when releasing the pistol by releasing the grip button, it automatically drops into the “holster” instead of on the ground. Thus, I will never lose the weapon and can always grab it again from the other side, although there is a cooldown before it is ready again after each time of being released.

Similarly, *Sairento VR* presents an efficient design of automatically “releasing” the sword to its back slot when switching to the pistol from a holster slot on either side of the waist. The player can grab the sword by moving either hand over either shoulder. Like in many VR games, the back or shoulder slots are used for longer weapons. What is special in *Sairento VR* is that the weapons are also bound to the player.

Figure 117. In *Sairento VR*, when the player grabs the pistol from the right holster with the sword at the right hand, the sword is automatically released to the back slot. Screenshots from https://www.youtube.com/watch?v=Mt5YxdmoX40.

With the two weapons and their respective slots from either shoulder and either side of the torso, the game implements the smooth transitions between switching the two weapons. The pistol also automatically reloads when placed into the holster with a reloading sound that takes a couple of seconds before it is ready to be taken out again. The difference in the side slots between this *Sairento VR* example and the *STRIDE* one is that in *Sairento VR,* the player can take the pistol from either side of the body, but in *STRIDE,* one hand can only take a pistol from its opposite side. Besides, *Sairento VR* does not release the weapon when the player releases the grip button except when the control is done on the slot.

Half-Life: Alyx also features a two-slot solution, but instead of having a holster at the side near the chest or waist, it has two wrist slots, as shown in Figure 118. There are two benefits that I think reconcile the value tensions and a minor drawback in this case.

The first benefit is that the wrist slots avoid the visibility issue of other slots because the player controls the hand and wrist positions. Thus, the player can always see clearly where the slot is when storing an item in it. The second benefit is that each wrist slot can store multiple items, like three of them, at one time. The game simulates a hyper-realitic storing technique, as seen in the right image of Figure 118, to increase the efficiency of carrying multiple objects in the game. The objects that can be stored in the wrist slots are equipable and can be selected by opening an inventory interface, which we will discuss in the next section. In addition, players also collect ammo and a special currency for weapon supplies and enhancement in the game. Those objects are stored in the backpack by releasing the hand's grip button over the shoulder.

Figure 118. Storing a healing injector to the right wrist pocket in *Half-Life: Alyx*

The minor drawback of the wrist slot design is that the storing phase couples both hands. One hand holds the object grabbed from the scene, and another hand awaits for the object to be placed in the wrist slot. However, considering this as a tradeoff for the benefits, overall, the wrist design improves the efficiency while maintaining a relatively high simulation value through the visual and metaphorical design.

This example from *Legendary Tales* of directly equipping a glove grabbed from the game scene to the hand "slot" is what I consider the neatest one in this section. As shown in Figure 120, the glove makes the hand a special slot that shows both high values of simulation and efficiency for the equipping action in traditional RPGs. The only pity is that the appearance of the hand did not update to reflect the gear. The hand slot only served as the entrance to store the glove into the inventory.

Figure 119. A neat example of equipping a glove onto the hand slot in *Legendary Tales*

There are other design aspects of body slots, such as visual and textual information display shown in Figure 120, which feel it is a cliché to discuss them again, and they are less critical and not the focus of this section.

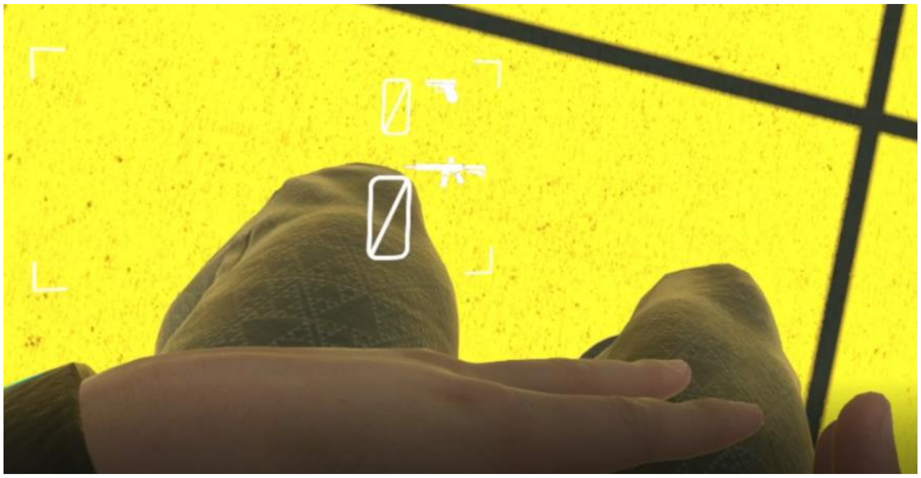


Figure 120. Visual and textual information (cliché) displayed on the body slot in *Boneworks*

5.2.2. Summary

To summarize this section on body slots, I group the examples by their slot and map their design values.

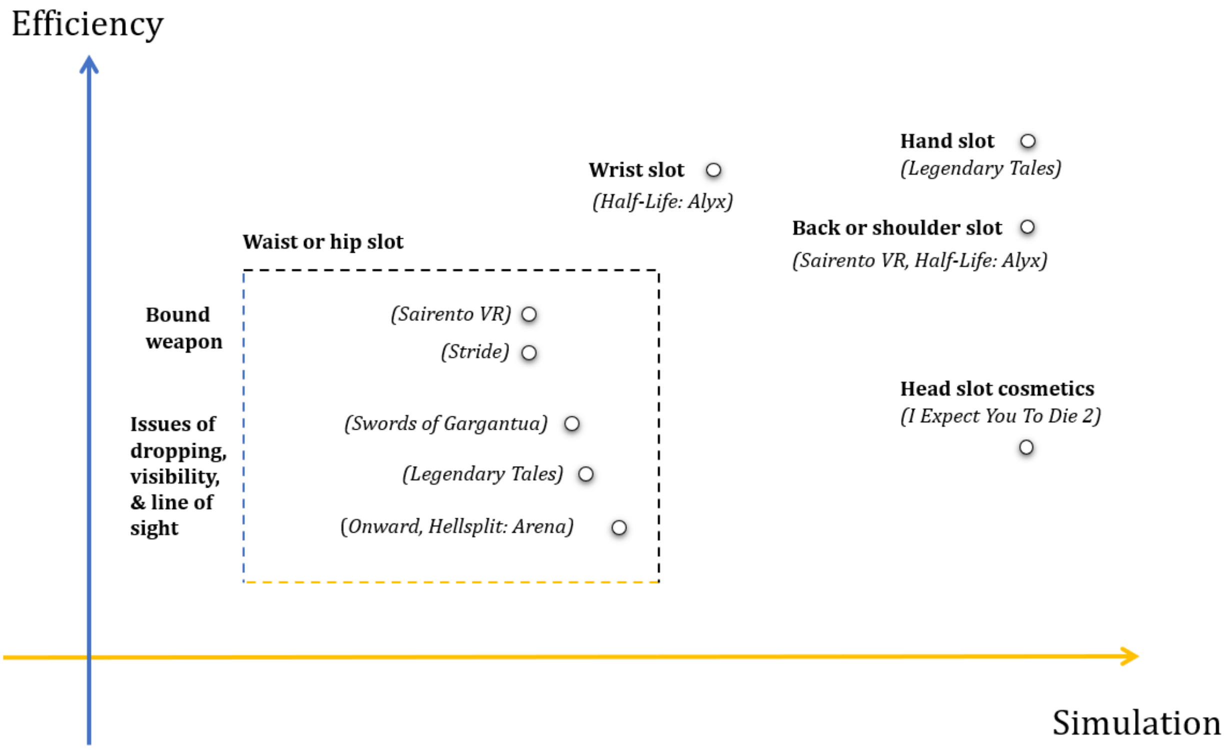


Figure 121. Summary and comparison of body slots for storing and equipping items

Overall, the waist or hip slot examples reveal more conflicts and limitations when judged by the design values than the other slots. They reveal the platform constraint of less accurately estimating their positions without knowing the player's body shape. Their disadvantages in the game contexts include issues of dropping, visually confirming the action, and line of sight. The two better solutions use special designs of bounding the weapon and reducing the number of slots. Another efficiency heuristics is not to overwhelm players with too many body slots, which can cause limitations in sizing and positioning them according to Fitts' law, not to mention the cognitive burden for remembering what items are in which slots.

The hand and head slots are easily high simulations because they are based on accurate position tracking and only deal with straightforward equipping interactions. The head example in the section is ranked with a lower efficiency only because it reduces sight and does not have functions related to game mechanics. However, I think they still can extend their uses in the future, such as magical lens effects. Regarding the hand slot, I think it has considerable potential for directly equipping hand gears and rings in future VR RPGs.

The back or shoulder slot is widely used for storing long weapons and the backpack metaphor. Also, accessing it is reasonably efficient by reaching either hand over either shoulder, as long as not dropping the item on the ground.

Finally, the wrist slot in *Half-Life: Alyx* further extends the storage capacity of a body slot by making a hyperreality wrist technology that fits the game's overall theme and supporting it with an inventory interface for selecting between multiple objects.

5.3. Inventory: Select and Switch

This section will review design examples of various inventory interactions for quickly selecting and switching items.

5.3.1. Examples

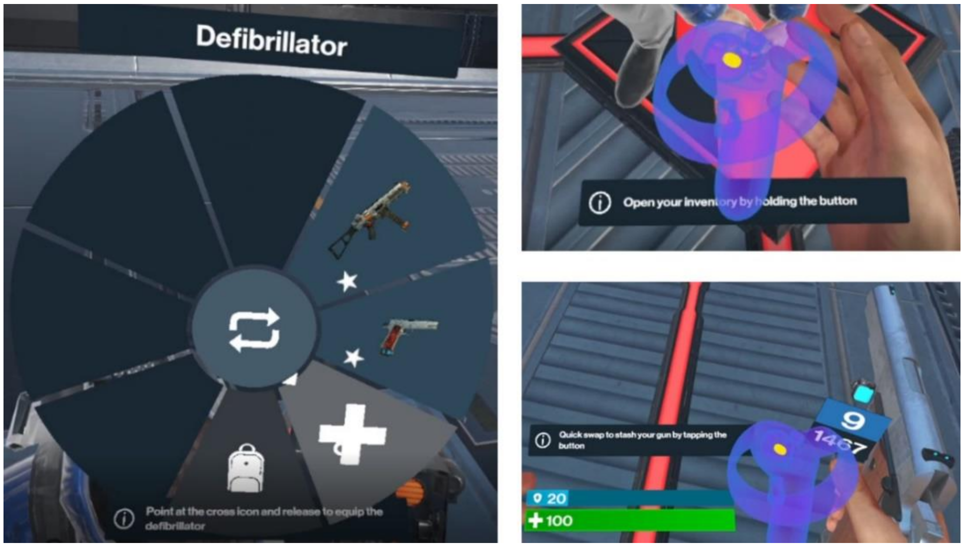
As a transition from the previous section, let us start with the inventory interface in *Half-Life: Alyx*, which builds on the wrist slot storage to select one of the items to equip.

Figure 122. The hand inventory interface in *Half-Life: Alyx* (left side) and its controls (right side)

In this example (Figure 122), the inventory interface is opened by holding the joystick button on the right (dominant) hand controller. By design, the inventory only contains a limited number[22] of items that can be laid out in hologram icons. The selection is made by directly touching one of the hologram icons with the hand. Additionally, the game implements a quick control to switch to the previous item by tapping the joystick button.

As we discussed, this design is efficient to avoid putting each object in a separate body slot. The holograms that are not being selected appear more transparent than the one that is being selected to avoid blocking the player's sight of the game scene. Moreover, the selection interaction strives for a high simulation value by using the holograms and matching their visual style to the overall artistic aesthetics of the game. The two controls show consistency in controller mapping. They also show the conventional metaphor and literacy of tapping for quick switches and holding for a detailed menu.

Using the same controls of tapping and holding, another well-received FPS game *Population: ONE* presents the inventory interface in a pie-menu style (Figure 123). The game also features the pointing control instead of the direct hand touching for selecting an item, which is more efficient for the pie menu that appears at a distance from the user.



Point at the cross icon and release to equip the defibrillator

Open your inventory by holding the button

Quick swap to stash your gun by tapping the button

Figure 123. The pie-menu inventory interface in *Population: ONE* (left side) and the controls of holding and tapping the right joystick button (right side).

---

[22] In later phases, the number of hologram icons increases as the player unlocks new weapons, and the extra ones are laid out on the left and right sides, forming a cross for players to pick one. What I meant by "design" also involves the game using the multi-tool to handle all kinds of puzzle objects to avoid the need of expanding the toolkit.

Regarding the design values for this example and how it is compared with the *Half-Life: Alyx*'s example, I find they are almost equal in terms of matching the game's overall visual style. However, I slightly favor the *Half-Life: Alyx*'s example for both simulation and efficiency values. Its visual design makes it narrativized interface, and the direct touch is slightly faster and easier than the pointing control for selection. On the other hand, the pie menu can stand out more efficiently for its larger capacity and consistent size in the long term.

Another example from *Budge Cuts 2* shows a similar design as the *Half-Life: Alyx*'s example, but the similarity is not in the controller mapping but how the inventory is based on the hands and the direct touch selection.

Figure 124. The handheld object select menu (left) and the inventory menu (right) in *Budge Cuts 2*

As shown in Figure 124, *Budge Cuts 2* also has a limited number of items and weapons, such as the translocator for teleporting and the bow, which rationalizes the direct-touch selection and the close layout of the menu that shows up above the controller. It simulates the robotic design of the character whose hands can equip various functional modules.

In addition, the inventory menu features four slots of stored objects and the "adding-to-the-inventory" interface (the blue bubble with a plus sign) right below them. Although it is a hyperreality technology, it is a decently efficient solution for storing four objects that can be quickly thrown to attack enemies and replacing any of them with a new object, all done on the same interface. The controller mappings for the handheld menu and the inventory menu are Y and X on the right-hand controller, respectively, simulating the relative positions of the two interfaces in the game.

Next, we will see two distinct inventory styles from VR RPGs.

The first style is what I would call a "3D inventory", featuring the example of a chest inventory with grids of storage space in the game *Shadow Legend VR* (Figure 125).

Figure 125. The 3D chest inventory and its use in trading with a merchant in *Shadow Legend VR*

This is perhaps the most realistic inventory that I have seen in VR games. It is technically a 3D chest with the word "Inventory" engraved under its lid. The character keeps and carries it through the game and can bring it out at any time. In detail, it simulates the grid storage space but also adds efficient features, such as stacking the same item in one grid and the seemingly infinitive space as indicated by the "page navigator". Although the detailed efficiency features are not realistic simulations of a physical chest, the sense of realness has already been set by the overall form of the chest.

Its use case of trading with a merchant adds to the value of simulation. When I put and opened the inventory on the merchant table, it was exciting to see what I had got so far, and the form of the chest and the embodied actions with it immersed me in the mood of trading. When I grabbed a golden cup in front of the merchant and raised my counter sell offer, the merchant reacted either madly or gently in his turn of bargaining depending on whether my offer was reasonable to him. After a few rounds, we agreed on the deal. To me, it was such a realistic trading experience that extended the interfaces in traditional RPGs into 3D VR space and added gamified bargaining dynamics to make the deal a little fuzzy.

The second style is the 2D menu interface in traditional RPGs. However, unlike the 3D inventory in the form of an object in the game, it feels non-diegetic, especially when it appears as a tilted canvas in the game world. As an example shows in Figure 126, the strength of such an interface is its efficiency in displaying game-related information at any scale.

Figure 126. The inventory page with character stats, equipment slots, and item information (left); the point-and-click interaction for viewing the item information and equipping it (right). *Legendary Tales*

5.3.2. Summary

Although inventory interfaces are usually mundane 2D menu interactions in traditional games, VR presents a novel and unique design direction using the 3D space. For this summary, I will use the dimension as a key criterion to evaluate the values of simulation and efficiency, as the controls are mostly button clicks except for direct touching.

As shown in the mapping in Figure 127, the two examples of the 3D and 2D RPG inventory interfaces set the respective highest and lowest values in simulation, and the hand-based interfaces sit in between. Overall, they achieve moderate to high-efficiency values due to the straightforward and independent nature of inventory interactions. The 2D menu is given the highest efficiency value for its textual and panel display of related game information. However, the 3D ones often incorporate 2D displays and take advantage of the space near the hands and direct touch and grab interactions, making them efficient for selecting and switching items in various specific game contexts.

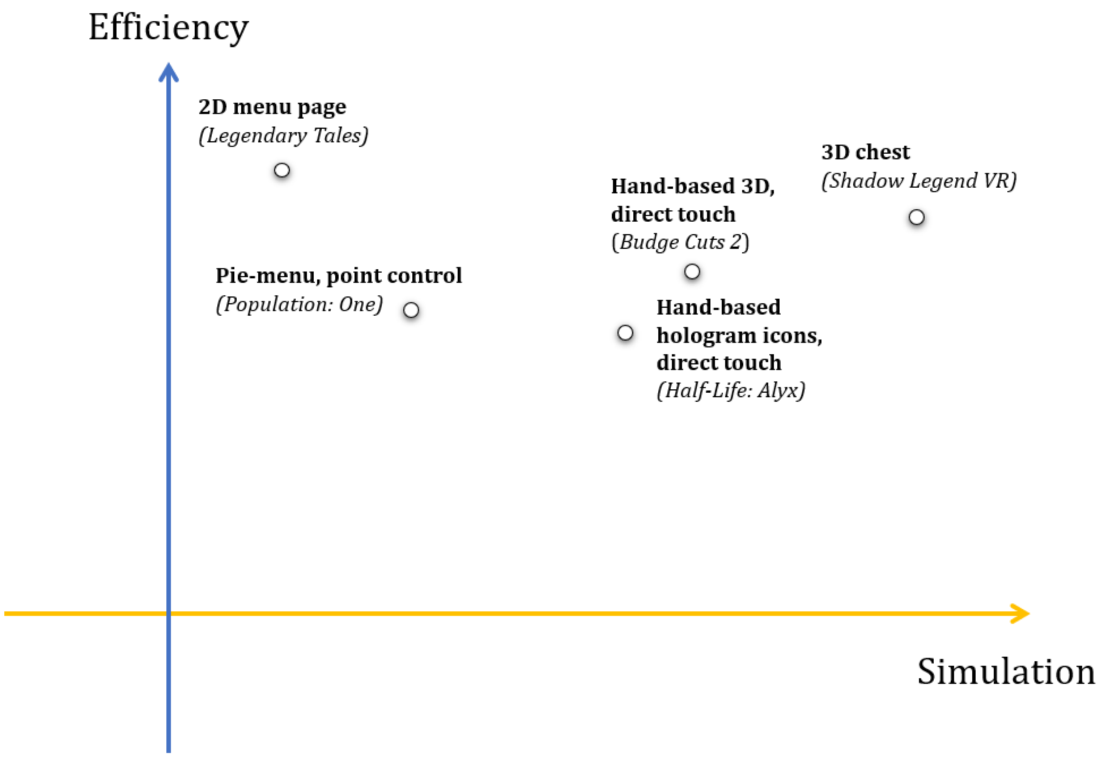


Figure 127. Summary and comparison of inventory interactions

6. FPS Combat Interactions

So far, we have discussed locomotion and object interactions. They set the stage for players to engage and fight in combats in the gameplay. In the following two sections, we will see how various combat interactions incorporate and balance the design values of simulation and efficiency.

This chapter will review FPS combat interactions around shooting, including aiming, firing, being hit and dying, reloading, throwing a grenade, and interactions with map and signaling. Chapter 7 will review RPG combat interactions with cold weapons and abilities, including weapon swing, parry and block, dodge and dash, ranged attacks, and consume and recover.

Let us get ready to see some guns.

6.1. Aim

Overall, the aiming interaction design differs on the level of aids for aiming. The aid manifests in the forms of visuals, mechanics, and hand controls. Unlike how traditional FPS games use a 2D cursor at the screen center to facilitate aiming, VR FPS games feature a manual approach of holding the gun with the moving hand and aligning the aiming slots with the target. To better evaluate and compare the examples, I divide them based on the type of the gun into the groups of pistols, (normal) rifles, and sniper rifles.

6.1.1. Pistol

Pistols are usually held by one hand, and players aim by aligning their sight, the aiming slots and the target on the same line. A common visual aid for aiming is to highlight the aiming slots with a type of color, as shown in Figure 128. With this visual design, the aiming slots become a metaphor of the aiming cursor in the FPS literacy from traditional games. Therefore, their sizes and color contrasts affect the efficiency value. The simulation value is minorly affected because the highlight can be rationalized as a simulation of a real-world modern material technology applied to aid the aiming.

Figure 128. Visual highlights for the aiming slots of pistols. The green one in *Boneworks* (left) has better contrast with the environment than the white one does in *STRIDE* (right).

Half-Life: Alyx further extends the technological simulation of the game's future theme by adding a larger visual aid for aiming, as shown in Figure 129. The cylindrical hologram makes aiming easier because players no longer have to focus their vision onto the smaller aiming slots. The game design makes the aiming aid only available as an upgrade purchased by the resources players collect in the game.

Figure 129. The pistol’s visual aid for aiming in *Half-Life: Alyx*

The color-coded aiming slot presents a different account of simulation that is difficult to be explained by any realistic material technology. As shown in Figure 130, in the normal state, when the pistol is not aiming at any target, the aiming slots are white. They turn green when aiming at a valid target and red when out of ammo. The color codes improve the efficiency of knowing when the timing is accurate. However, they are not realistic simulations. They are at most using metaphors of what each color implies in our daily lives: green means valid or “good to go”, while red means the action is invalid or forbidden.

Figure 130. Color-coded aiming slot in *The Walking Dead Onslaught*: turns green when aiming at a target, turns red when out of ammo

Laser, or aiming cursor, of course, is not completely abandoned in VR. In fact, they are much more useful in VR than in traditional platforms to help the 3D aiming. As shown in Figure 131, the player does not have to align the gun in the same line formed by their eyes and the enemy. Instead, players can aim with the gun at any angle by twisting the wrist, significantly improving aiming efficiency.

Figure 131. Laser for aiming in the game *Frostpoint VR: Proving Grounds* (inXile entertainment, 2020) (left), aiming cursor in *Population: ONE* (right).

The game *Pistol Whip* implements an “aim-assist” mechanic in its gameplay. As shown in Figure 132, I only had to point in a rough direction to hit the target. This design largely increases the efficiency value at the cost of simulating aiming mechanically. It serves to keep players in the flow state of shooting by lowing the challenges of aiming, which otherwise can be difficult with the character running forward automatically and encountering enemies that appear in all angles.

Figure 132. The aim assist mechanic in *Pistol Whip*'s gameplay allows the player to hit targets by pointing to its rough direction.

6.1.2. Rifle

As for rifles, the aiming techniques and design variations are mostly the same as those discussed in the pistol examples. The *Onward* example also shows a purely mechanical aiming with no visual aids. However, what comes as an additional major design variable to the rifles is holding them with both hands.

Figure 133. Two examples of holding a rifle with both hands (left: *Pavlov VR*, right: *Onward*).

Many VR FPS games simulate the physical effect of recoil and the benefit of holding a rifle with both hands to increase stability. It almost becomes necessary for aiming because only in the two-hand configuration will the gun stabilizes in the position straight in front of the eyes. This design exchanges the efficiency of using another hand for a more excellent simulation, but then the simulation also increases the aiming efficiency in terms of stability.

Even for rifles using the aiming cursor, the example in Figure 134 shows how the simulated recoil destabilizes the cursor aiming with only one hand.

Figure 134. With only one hand holding the rifle, the recoil mechanic becomes dominant. The efficiency brought by the cursor does not stabilize aim, even in a short-range, as shown in an example from *Population: ONE*

6.1.3. Sniper Rifle

Sniper rifles typically simulate aiming with the scope. The player looks close into the scope to enter the scoped aiming mode. Overall, there are two visual design variations for scoped aiming.

The first visual design is like the "tunnel vision" for movement and has the same disadvantage of losing the broader sight of the game scene (Figure 135). It simulates viewing through a scope; however, I prefer seeing the full amplified view rather than the outer part in the black screen. Worse still, in every moment of firing, I would lose entire sight due to the simulation of the spark.

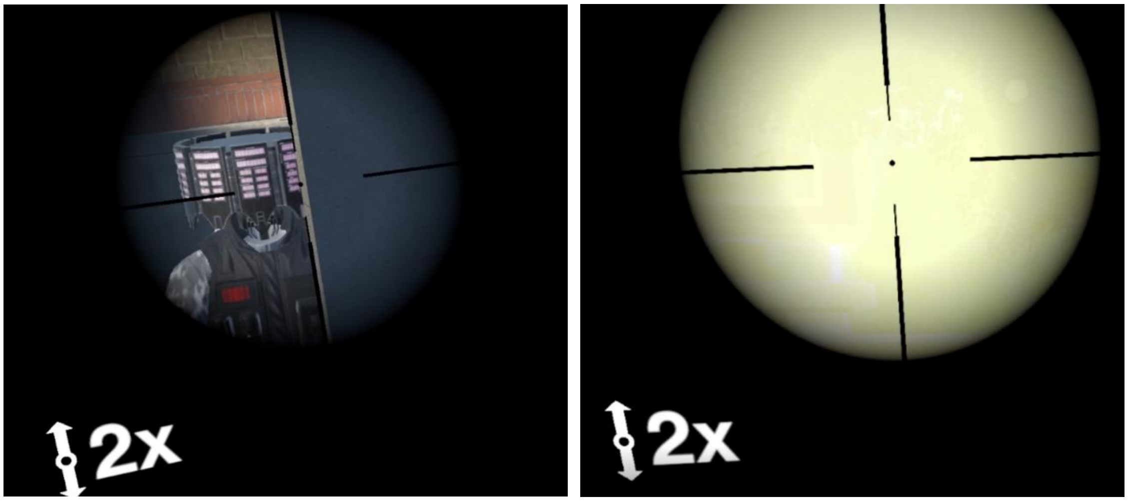


Figure 135. the “tunnel vision” style of scoped aiming and the spark when firing the gun *in Population: ONE*

The other style of scoped aiming is shown in Figure 136. It simulates the aiming through the scope without being fully immersed by the scope. Perhaps we can call it “third-person scoping” to differentiate from the “first-person scoping” in the first example. The eyes are still outside the scope, with a good distance to see the game scene. This design reconciles the value tension in the first example by combining the amplification benefit with the sight benefit, achieving high simulation and efficiency values.

Figure 136. *Pavlov*’s “third-person” scoped aiming with the amplified view inside the scope and the normal sight outside

6.1.4. Summary

As shown in Figure 137, overall, the visual design plays a less significant role than the mechanics and control techniques in determining both values for aiming interactions. The purely mechanical aiming technique without visual aid achieves a high realness in the simulation but relatively low efficiency. The two-hand and scoped aiming for rifles and sniper rifles bring extra efficiency because of their mechanics and excel in simulation. The tunnel vision scoped aiming, in my experience, limited the aiming efficiency by having a black screen around the amplified view within the scope, which may also be a partial simulation of the real-world referent. The aiming-assist mechanic shows a tradeoff. It lowers the required accuracy of the embodied, hand-tracking-based VR aiming, but at the same time, allows an inaccurate simulation of aiming in the real world.

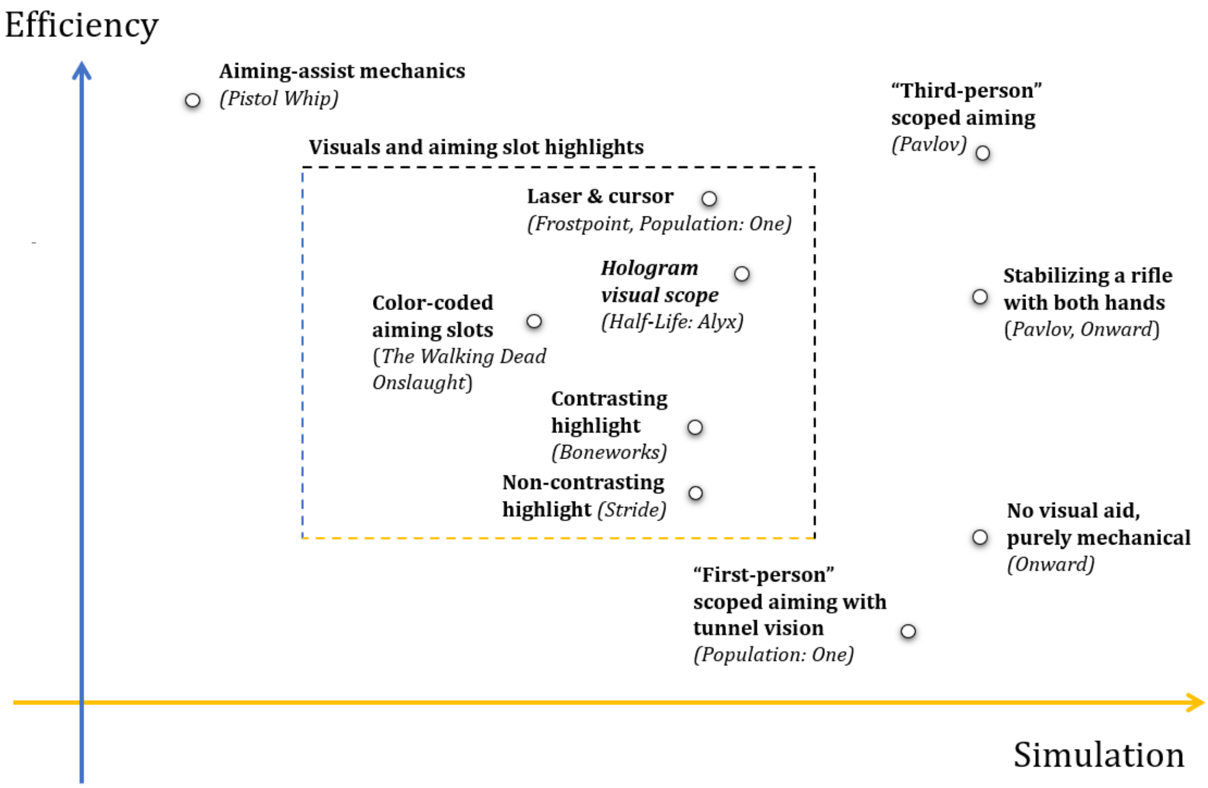


Figure 137. Summary and comparison of aiming interaction examples

Within the group of visuals and aiming slot highlights, first, the color makes a difference for the aiming slot highlight. The color codes score higher on the efficiency for its functions but less on the simulation value than the constant color design, which can be rationalized as a material property. Similarly, the hologram aiming scope and the laser and aiming cursor visuals simulate more advanced technology that facilitates aiming. Therefore, we can conclude that simulating aiming-assistive technology as visuals can reveal and reconcile the value tensions for this interaction design.

6.2. Fire, Hit and Die

This section will discuss design cases on the gameplay dynamics during firing, being hit and dying in VR FPS games. Like aiming, they also heavily depend on visual forms and feedback. However, many shooting-related dynamics and visuals in VR games are legacy from traditional games, including floating damage numbers, realistic weapon firing visual effects, bloody splashes on the screen, recoil effects, scoped aiming, and sometimes even distorted bodies. So instead, I want to focus on unique design cases that significantly impact the VR gaming platform and its embodied interaction experience.

Specifically, I identified two categories of design cases – projectile trace and dying visuals, where the variations in visual forms reveal and amplify the value tensions and affect comfortableness in the head-mounted display during the firing, being hit, and dying interactions.

6.2.1. Projectile Traces

VR FPS games tend to amplify the visual effects of projectile traces. Some games connect the salient visuals and bullet speed to time-slow mechanics and allow players to react to income bullets and dodge them. *Superhot VR* (Superhot Team, 2016) was one of the early successful VR FPS games that featured projectile trace and a time control mechanic, and it probably influenced the design of VR FPS games to follow. Some games do not have a time-related mechanic, but the projectile visuals are still there to intensify the sense of presence in the first-person immersive VR perspective.

I find emerging design approaches of projectile trace reveal and amplify the value tensions between simulation and efficiency. I experienced a lot of uncomfortableness with these visuals, from where I started to believe that any uncomfortable experience leads to inefficiency.

For example, in *Sairento VR*, the bullet traces are visualized in distinct orange lines as they fly towards the player from different angles. As shown in Figure 138, some traces are straightly hitting the eyes or passing through the space very close to the player.

Figure 138. Bullet traces hitting the player in the game *Sairento VR*

The intensity can be easily scaled up when many dense lines are incoming from multiple enemies or when a sharp blade wave is incoming anytime during the fight, as shown in Figure 139.

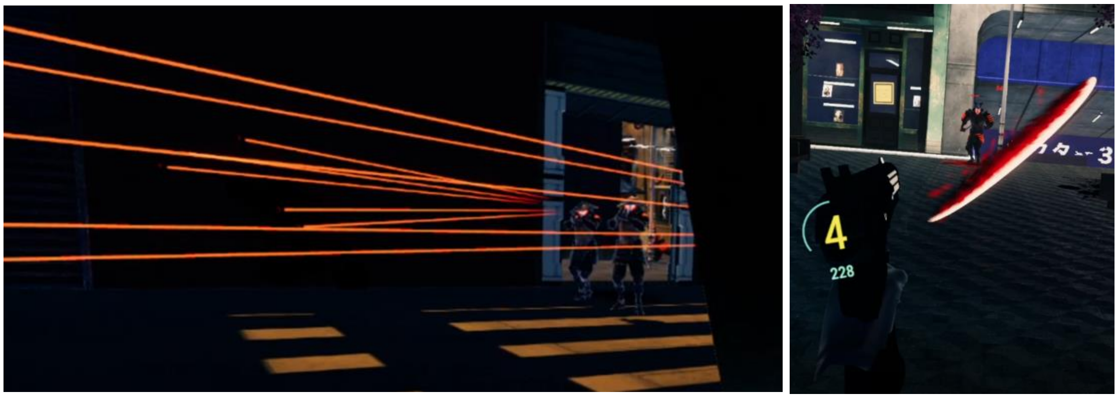


Figure 139. Too many dense lines (left) and a sharp blade wave (right) in *Sairento VR*

Worse still, when the player is in low health, the game blurs the screen to simulate the blurring and distorted sight when injured and losing consciousness, and the incoming projectile traces look more dangerous and despairing (Figure 140).

Figure 140. Blurred screen when the player is in low health in *Sairento VR*

In *Pistol Whip*, although the bullets are more in a cartoony and polygon style, their visuals, including the big red dot signifying where they come from, the circles on their traces, and the large white light effect behind the bullet, all intensify the experience of seeing and dodging them.

Figure 141. Amplified bullet traces in *Pistol Whip*

When being hit in *Pistol Whip*, the field of view is splashed with red sharp triangle glasses to simulate blood and a feeling of being hurt. This momentarily reduces a good amount of the field of view and greatly intensifies bad feelings of being hurt and bleeding.

Figure 142. Visual effects when being hit in *Pistol Whip*

The example in Figure 143 from *I Expect You To Die 2* shows that even the scene looks normal and does not have a typical FPS setting, a remote laser beam directly aiming at the player through a small gap can still arouse sickness. In addition, the game design also involves thriller content that further intensifies the experience.

Figure 143. The red laser beam from a remote assassinator aiming at the player in *I Expect You To Die 2* (in VR, the laser beam was more straight into my eyes. The screen recording has a bit of distortion and was hard to capture the direction)

In summary, I think the projectile traces, especially the bullets, do not simulate the physics of their speed but overly amplify their visuals and the effect of "shooting at players". They can be efficient in signifying the player to react and dodge in real-time to suit the game design. However, even for this reason, I still find many examples like these that make me sick every time the projectile traces hit me. I am a player who is immune to motion sickness most of the time in VR games, but the sharp, fast incoming projectiles that directly point to my eyes intensify the sickness. Because of this, I consider them less efficient in maintaining a friendly and immersive gameplay experience than the accurately simulated bullet speed and effects would do. As I stated, any uncomfortable interaction design in VR is inherently inefficient.

6.2.2. Dying Visuals

Following the trend of projectile traces during the combat, some VR FPS games, ironically, keep the project traces static (poor simulation) as the dying visuals. Dying is perhaps the most negative thing that a player wants to experience in the gameplay. However, the projectile traces that directly pass through the player's view and stay in the view makes the experience even more painful.

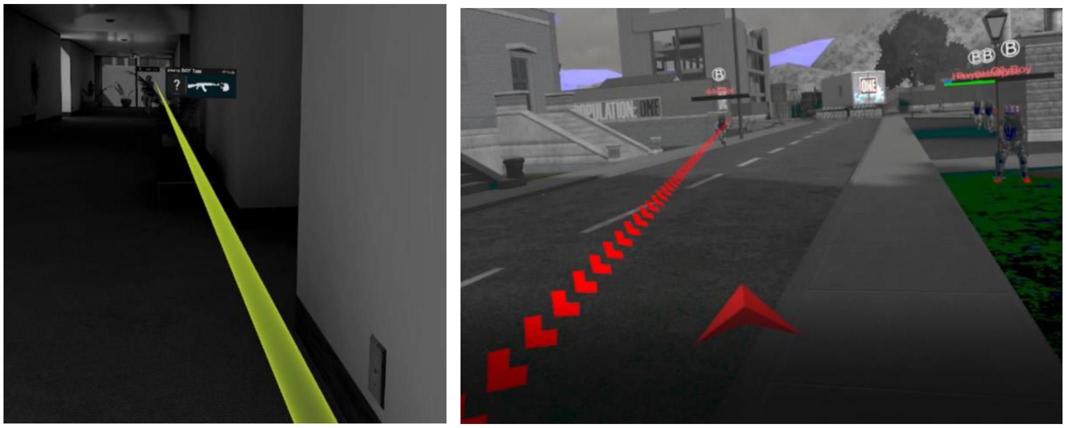


Figure 144. "Projectile trace" style dying visuals in *Pavlov VR* (left) and *Population: ONE* (right). In VR, the lines are more directly through the eyes or closer to them than in the screenshots.

The function of this "projectile trace" style dying visual (Figure 144) is obvious – it shows the trace of how a mortal bullet penetrates the player and displays UIs of the killing information. However, this simulation, or functional narrativization, does not bring the value of efficiency to the player if it makes the player uncomfortable. It does not simulate the instantaneousness of dying and its negative feeling. Dying should be quick instead of leaving the player to experience the pain by watching an uncomfortable line going through the camera. In VR, where the player is deeply immersed in the headset, the project trace dying visual can be more painful than how it would be in traditional screen-based platforms.

Another inefficient design related to dying is the inability to move the camera. This is involved in the project trace problem but can also lead to more problems. For example, in Figure 145, the enemy character can point their rifle closely to the camera (the player' eyes) after they killed the player. When I experienced this moment, all I was doing was moving my head away to avoid looking at the gun. However, I wish I could have controlled my camera to move in a ghost form or quickly skip watching the enemy step on me.

Figure 145. An enemy character walks into the dead player and points the gun closely towards the camera in *Onward*.

Another despairing "interaction" design example is the "give up" option after being critically injured and knocked down, as shown in Figure 146. In many team-based FPS games, being knocked down is a state where players can wait and hope for rescue from teammates. But it also limits their movement and makes them vulnerable to the enemy. Considering the aforementioned experience of facing enemies and their bullets, I think the give up option gives the player the option to avoid uncomfortable situations. Thus, in theory, it also achieves high simulation and efficiency values.

Figure 146. The "give up" option after being critically injured and knocked down in *Frostpoint VR: Proving Grounds*

Besides the projectile trace style, other common practices of the dying visual design include the filtering or fading effects and the tunnel vision style, which are often seen in the RPG game genre. For example, as shown in Figure 147, the *GORN* example uses a closing eye effect to simulate the process of losing consciousness and dying. The game *I Expect You To Die 2* uses a similar tunnel vision effect to fade out the game scene.

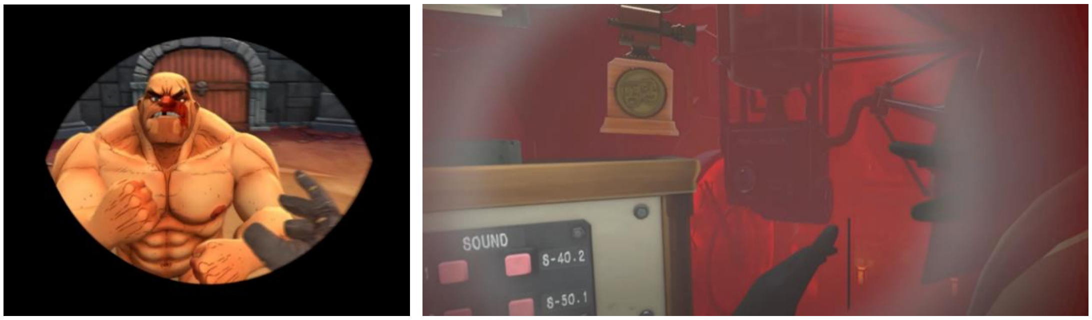


Figure 147. Dying visuals with tunnel vision effects in *GORN* (left)'s "closing eye" style and in *I Expect You To Die 2* (right)'s fading style

These two visual design examples are relatively more comfortable to watch in VR, but the following filter effect in Figure 148 still builds up high pressure with the red color.

Figure 148. The "bloody filter" effect of dying in *Hellsplit: Arena*

In summary, the comfort level is the most important design factor that affects the player experience of dying. Although the projectile trace style has the function of displaying kill information, it potentially causes sickness and breaks the experience. I insist that any uncomfortable design in VR leads to low efficiency, as it prevents players from continuing to play the game, no matter how well they simulate various aspects of the firing, being hit, and dying and how functionally efficient they are in the game context. The filter, fading out, and tunnel vision can mitigate the sickness to some extent. Still, as a player I prefer having control of my dying camera and options to customize the dying visuals.

6.3. Reload

Like how aiming gets reworked in VR, reloading is also implemented with body actions in most VR FPS games.

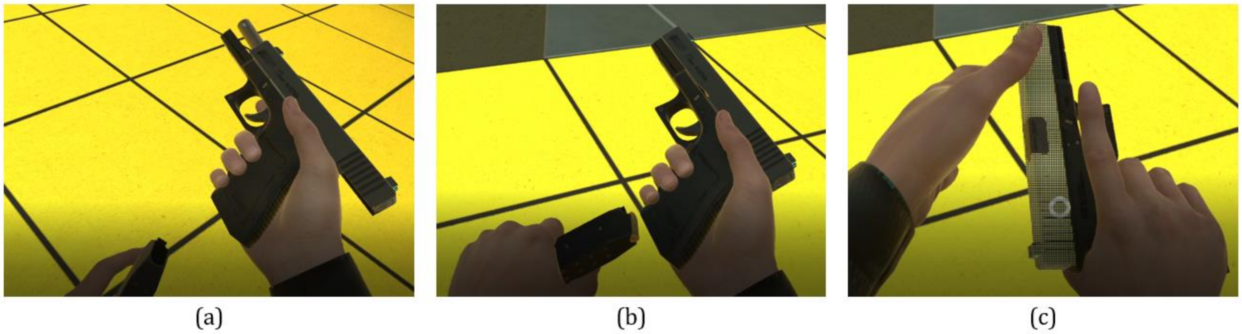


Figure 149. Reloading steps in *Boneworks* as a standard practice: (a) pull the magazine out of the gun; (b) grab a new magazine from the waist and insert it into the gun; (c) pull the slider on the gun to chamber round

As demonstrated in Figure 149, when the gun is out of ammo, the player has to (a) pull the magazine out of the gun, (b) grab a new one from the waist slot and insert it into the gun, and (c) pull the slider on the gun to reload. These steps all have high interaction fidelity and reproduce the physical behavior of reloading. However, performing all the steps can take a considerably long time, like about three seconds in combat. However, there are a few techniques to simplify the manual process.

The first technique is to use the snap mechanic in step (b) when they insert the magazine into the gun, instead of finding the perfect angle and moving the hand all the way continuously and strictly in a line. This increases efficiency but does not reduce stimulation value because the hand moving to the general position of the slot is enough to reproduce the meaning of the step.

Another technique we can see in this example is the visual highlight of the slider to signify the pulling action. As discussed in (Krompiec & Park, 2019), the pulling direction can be enhanced by mapping the direction of hand movement onto that of the slider to ease the action.

Besides these techniques, we will look at other design cases that simplify or problematize the reloading action and evaluate how they balance the simulation value.

6.3.1. Controls and Visuals

Figure 150. Options to press the A and B buttons to eject the magazine and chamber round, respectively, in *Half-Life: Alyx*

As shown in Figure 150, in *Half-Life: Alyx*, the control mappings offer alternative interaction techniques to perform the steps (a) and (c) – ejecting the magazine and chambering around. The positions of these two buttons on the right-hand motion controller map well onto the positions of the parts on the gun, with the lower A button connecting to a switch controlling the lower magazine and the higher B button connecting to an imagined “reloading trigger”. These controls simplify 2/3 of the manual actions while still maintain a relatively high level of simulation. In addition, when the gun is out of ammo, the lower part of the gun turns red, providing extra visual hints for the player to reload.

On the note of the visual hint for roading, the example in *Population: ONE* in Figure 151 explicitly shows the number of ammo left in the gun/inventory in the X/Y fraction UI layout. It works with the option to reload early to allow players to make strategic reloading decisions at any time in the game. It follows the “Visibility of System Status” usability heuristic for UI design (Nielsen, 1994) and increases the efficiency of the reloading by providing the player with extra context and agency. The UI display of the ammo is not a realistic simulation, but, in this case, the benefit of the efficiency is reasonably prioritized.

Figure 151. *Population: ONE's* X/Y fraction style UI to show the ammo left in the gun/inventory and the option to reload early

6.3.2. Mechanical Structures and Constraints

Besides the visuals and controls, the gun's mechanical structure and constraints involving the slider/handle also affect the design values within the reloading interaction.

For example, as shown in Figure 152, the handle (highlighted in yellow) needs to be pulled through the end to trigger the chambering successfully. However, the way I held the sniper rifle to pull the handle (and I see other people do this, too) was to use the left hand to reach the right side. Unfortunately, the sniper rifle has a long body, and it was hard to see the end position of the handle if I held it in this way. As a result, my attempt to pull the handle failed as it bounced back to the original position, and I wasted time and had to start over again.

Figure 152. A failed attempt of reloading a sniper rifle when the handle bounced back because it was not pulled through the end, *Population: ONE*

I checked the images of the real-world version of this weapon (AWP sniper rifle), and the game accurately simulates the mechanical structure of having the hand on the right side. However, its use cases are probably not the same between the real world and the gameplay. I suspect that in the real world, snipers use their right hand to pull the handle. It also has a stand to set it above the ground for snipping. However, in the game, the gun is used while moving and shooting at enemies coming from different angles.

Moreover, the reloading process typically uses the left hand to pull the magazine out and grab a new one to insert. In the example from *Pavlov VR* (Figure 153), the player has to technically put a large bullet into the slot after pulling the handle to reload. These steps all use the left hand, supposing the right hand keeps holding the gun (otherwise, it will drop).

Therefore, although the game accurately simulates the mechanical structure of these rifles, it does not fully simulate how the hands manipulate them and the context of using the weapon. For example, the games need to enable the left hand to hold the gun so that the right hand can pull the handle to ease this action. However, if the right hand is the hand to pull the handle, should it also be the hand to grab the ammo? Maybe. There can be some inherent constraints, such as one hand has to hold the gun, due to which it is still not possible to fully reconcile the tension between simulation and efficiency.

Figure 153. Reloading a heavy sniper rifle by inserting a bullet. *Pavlov*

Compared with this special case, other guns with different shapes and mechanical structures can better balance the two design values. For example, in Figure 154, for a normal rifle, as long as the slider for chambering round is not at the right side, pulling it with the left hand is fairly efficient.

Figure 154. Reloading a rifle with the left hand easily pulling the handle for chambering round at the center of the gun body behind the aiming slot. *Boneworks*

For revolvers, the game *Hot Dogs, Horseshoes & Hand Grenades (H3)* (RUST LTD., 2016) implements a clever reloading technique based on the weapon's unique mechanical structure. As shown in Figure 155, players can flick the controller holding the revolver to virtually exert an impulse force that opens or closes the cylinder that contains ammo. This flicking control achieves both high values of simulation and efficiency.

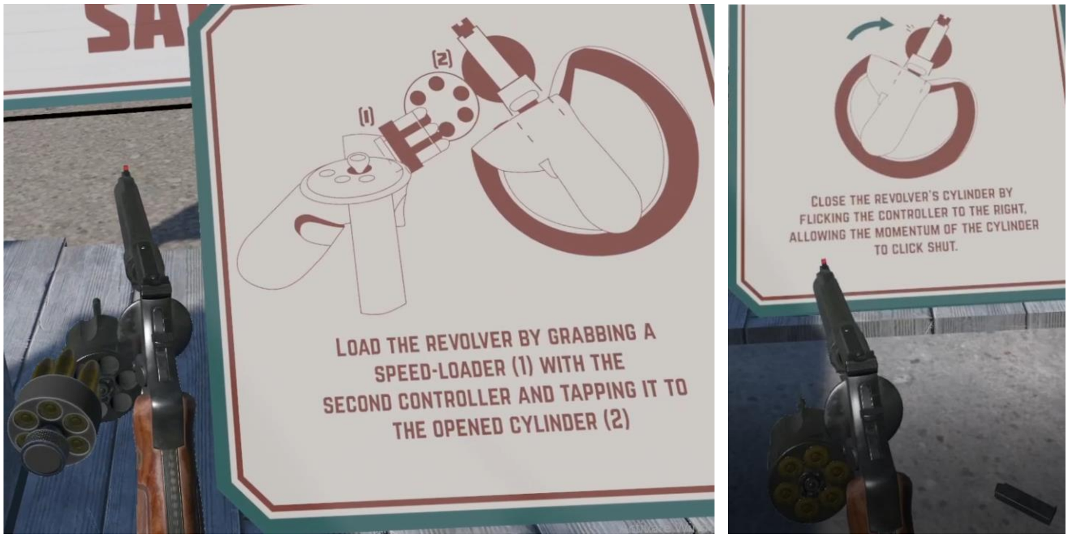


Figure 155. Reloading a revolver by flicking the controller to open and close the cylinder in the game *Hot Dogs, Horseshoes & Hand Grenades (H3)*

For reloading a rocket gun, *H3* implements a straightforward reloading method of directly inserting a rocket into the gun, as shown in Figure 156. It also uses the snapping technique to improve efficiency, as it is hard to keep the perfect straight direction when inserting the rocket without realistic haptics feedback and materiality.

Figure 156. Reloading a rocket gun in *H3*

As shown in Figure 155, the pointing down reloading is a faster technique than the standard manual approach as tested to take less time in Krompiec & Park (2019)'s study. It suits the fast-pacing gameplay in games such as *Pistol Whip* and *STRIDE*. Like the "snap" mechanic, this approach's efficiency builds on the loss of details in the simulation of the body movements and steps. Finally, another reloading option in this game is the "pistol whip" action of melee attacking an enemy with the pistol. It has a lower simulation value, and its efficiency depends on the timing of approaching the enemy. Therefore, it is only a situational choice but can still be a fun technique for this one game.

Figure 157. Fast reloading by pointing down the pistol. *Pistol Whip*

6.3.3. Summary

In summary, the reloading interaction reveals the value tensions between simulation and efficiency in terms of controls, visuals, and the weapon's mechanical structure and constraints. Overall, the visuals have a relatively smaller impact on the values than the other two.

As shown in the framework in Figure 158, the standard 3-step manual approach for pistols and rifles, the direct manipulation for the rocket gun, and the flicking control for revolvers, all achieve the highest simulation value because they have a high interaction fidelity and reproduce each step of the physical behavior. The sniper rifle examples show how the mechanical constraints of the gun lead to inconvenient and unnatural reloading, in which the left hand has to pull and handle on the right side of the weapon. The visual and control variations from the standard manual reloading improve efficiency but deviate from being realistic. Finally, the pointing down method is faster in time than the multi-step approach by abstracting the real-world behavior. The melee enemy approach is a situational fun technique but, in theory, scores poorly in both values.

Some of the aspects in visuals, controls, and mechanical structures can be combined to suit the specific gameplay style and design goals. Notably, the reloading speed is an important variable to keep in mind for designers, as it can significantly affect the game dynamics. For example, in my experience with the manual reloading that took a long time, I tended to "camp" or stay behind a cover to shoot enemies from a distance rather than be more radical and active to move forward in the open area.

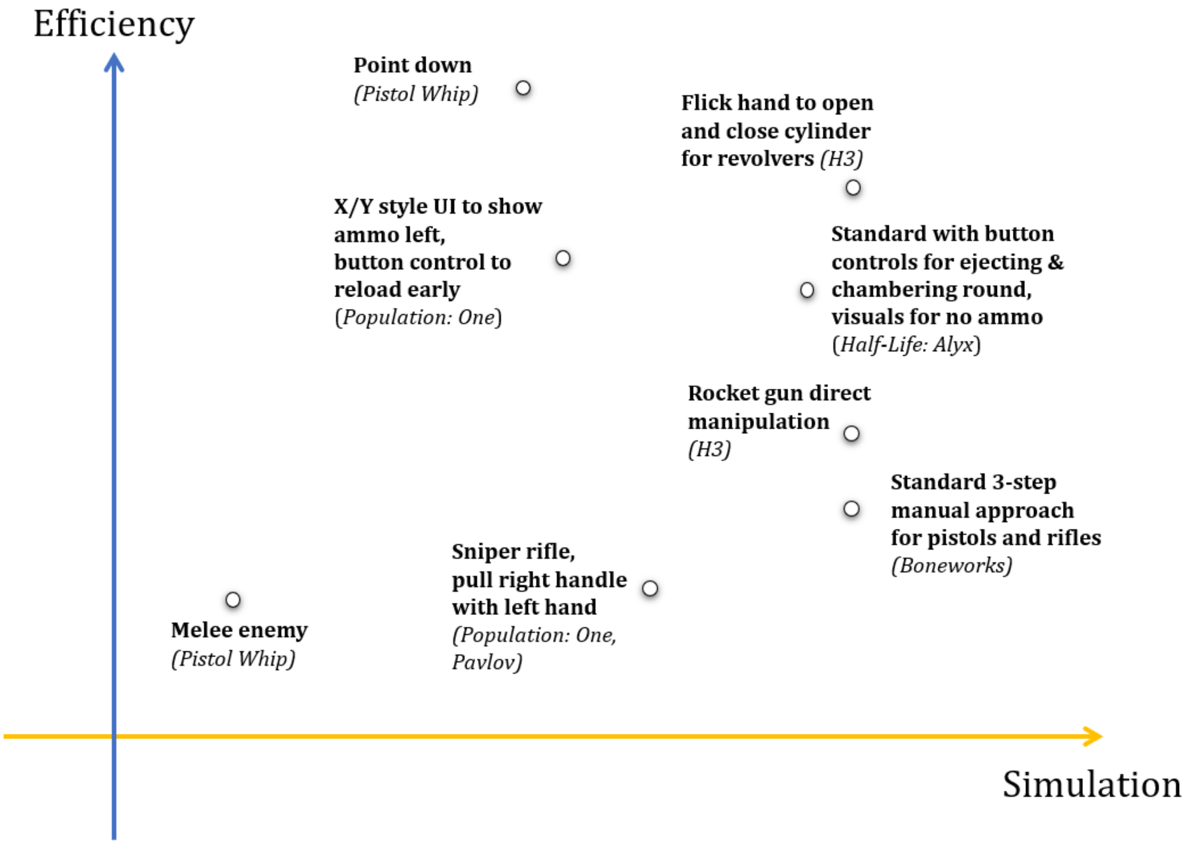


Figure 158. Reloading interactions: summary and comparison based on the simulation and efficiency framework

6.4. Grenade

Grenades are commonly used weapons in FPS games, and the grenade interaction reworks in VR with bodily actions. The design values within the interaction of throwing a grenade vary mostly with control techniques. In this section, we will discuss three different techniques of grenading.

The first is the realistic throwing technique that typically consists of 3 steps, as shown in Figure 159: (a) grab and pull the pin out by holding the trigger, (b) begin cooking the grenade by pressing the trigger (this step can be optional in some games), (c) throw the grenade by releasing the grip while throwing.

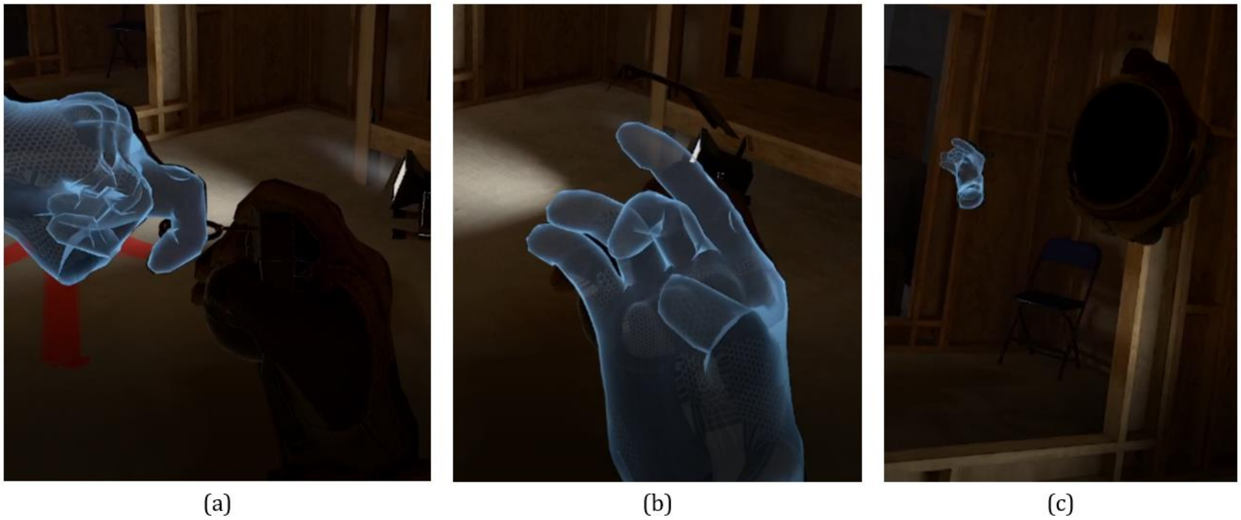


Figure 159. The realistic grenade throwing in *Pavlov*: (a) grab and pull the pin out by holding the trigger, (b) begin cooking the grenade by pressing the trigger (this step can be optional in some games), and (c) throw the grenade by releasing the grip while throwing.

Like the 3-step reloading interaction technique, this technique has a high simulation value because it reproduces the realistic steps and has a high level of interaction fidelity. The parts where efficiency may fluctuate between steps (b) and (c). Once the grenade begins cooking, players do not know when exactly it will explode. They may have an estimation such as in five seconds. While this is an accurate simulation of the real-world grenade, players can accidentally kill themselves when they hold it too long. Also, although the throwing step (c) looks realistic and has a high interaction fidelity, it still feels different from the physical throwing. In my experience, it was difficult to control the accurate landing position, making the action inefficient in achieving my goal in the game.

The second technique is the “aiming-curve” technique shown in Figure 160. This technique significantly improves efficiency by simplifying multiple controls into a button press to launch the grenade and providing the curve to visualize its exact landing position. However, it does not simulate the bodily throwing action.

The third technique is technically not counted as throwing a grenade. As shown in Figure 161, it uses a telekinesis technique to push the grenade towards an enemy. It looks like a reverse effect of the remote grabbing interaction. It further extends in the efficiency design dimension by turning the curve trajectory into a straight line, but ignores the simulation aspect.

Figure 160. Grenading with an aiming curve in the game *Population: ONE*

Figure 161. the telekinesis style of “throwing” a grenade in *I Expect You To Die 2*

6.5. Map and Signal

Map and signal interactions are commonly implemented features in FPS games for team communication. Since the VR platform has limited controls and promotes embodied actions, VR FPS games have explored the following techniques for map and signal interactions.

6.5.1. Tablets and Body-based Maps

The real-time combat feature in FPS games requires the map to be quickly accessible without pausing the game or pulling the player out of the game world to view a menu window. Therefore, an emerging approach for checking the map is to have the map displayed on a tablet in a body slot or anchor the map display with the body.

For example, in *Onward*, the player can grab the tablet to check the top-down view of the area on a small scale, as shown in Figure 162.

This tablet will show a coloured dot for any friendly soldiers still alive on your team. Green is full health, yellow is injured and red is critically wounded. A red cross indicates a soldier that has been downed but is available for a revive using your syringe. The objective will show on your map as a large white dot or symbol.

Please use the tablet to find and move towards the objective.

Figure 162. The tablet map interface in *Onward*

This tablet interface is a good simulation of high-end military technology with satellite support. In terms of efficiency, the tablet provides extra functions such as hacking a bomb to complete the objective in *Onward*. The only problem is that the tablet takes a body slot and can cause the inefficiency of mistakenly grabbing an item, as discussed in the object interaction chapter.

The tablet map in the game *Tales of Glory 2 – Retaliation* (BlackTale Games, 2020) also simulates the real-world tablet by exploiting its screen interface affordances for various commands. For example, it allows players to set waypoints for their squads and call supports to attack the area (Figure 163).


Now click on Air Support and watch the show...

Figure 163. The tablet map interface in *Tales of Glory 2 – Retaliation*

Other maps solve the body slot inefficiency by simulating a hyper-realistic hologram map anchored with the body. For example, in *Population: ONE*, players can press and hold the left trigger to view the map in front of them. As shown in Figure 164 (left), the game cleverly lays the map at an angle that does not affect the player's sight of the environment. Compared with the tablet interface, this design improves the efficiency and uses the hologram visual style to keep the simulation value after dropping the materiality of the map. In the game *STRIDE*, players can flick their right hand like Figure 164 (right) to quickly view navigation (shown in the arrow) and the objective. In this case, the map takes even less of the player's field of view, but the hand interaction can still take extra attention and agency away from the player during the focused parkour race, as the hand-swing is used in jumping and climbing controls.

Figure 164. Body-based hologram maps. *Population: ONE* (left) *and STRIDE* (right)

6.5.2. Radio Orders

This technique convinces players that they have a radio device at their ear to give orders. They can point their hand (usually the left hand assuming the right is holding the weapon) near the ear to activate the radio (Figure 165).

radio

Put your hand near your ear and hold trigger to activate the radio

Figure 165. Controls to activate the radio in *Pavlov VR*

In some games, this action is like the push-to-talk microphone control, and the radio becomes a narrativized interface for the voice chat. Other game design emphasizes the radio in the game context, such as a device to send tactical commands to control the AI teammates. For example, the FPS game *Tales of Glory 2 – Retaliation* features crew missions coordinated by radio orders. The player has several preset options to order the moves of the crew.

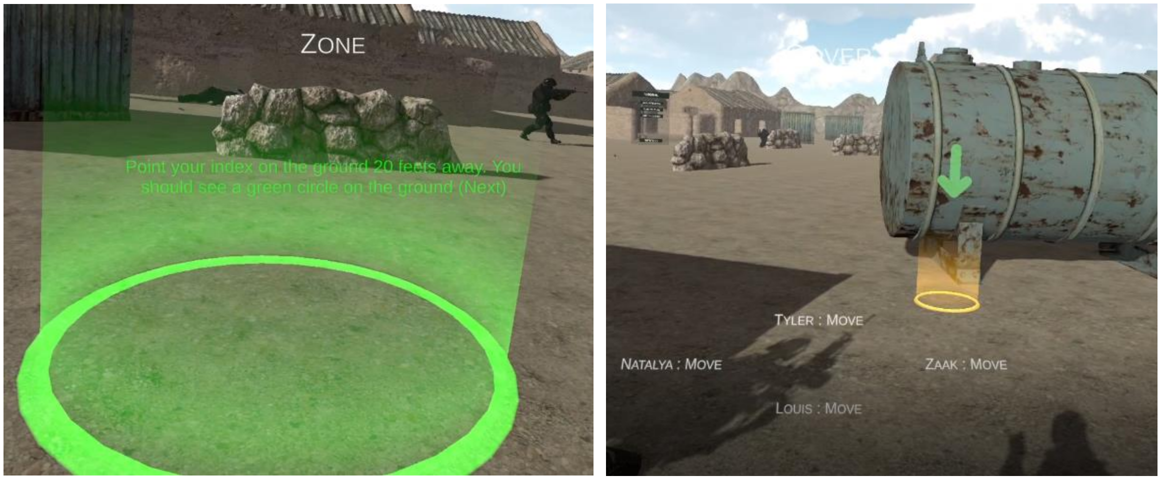

Now, move your left controller stick up to give the Search & Destroy order. One of your squad member will acknowledge it (Next)

Figure 166. Preset options in the radio command system in the game *Tales of Glory 2 – Retaliation*

As shown in Figure 166, after the radio is activated, the player can move the left control to select one of the four preset commands displayed in a text layout at the center of the screen. Besides, if the player activates the radio by moving the left hand to the ear but not press the trigger, they can wait to see the names of the four members in the same layout and select one of them to perform one of their specific actions in the same layout. However, this interface feels too specific for this game design with four members in the squad and four preset commands, which map to the four directions of the joystick.

6.5.3. Pin the Environment

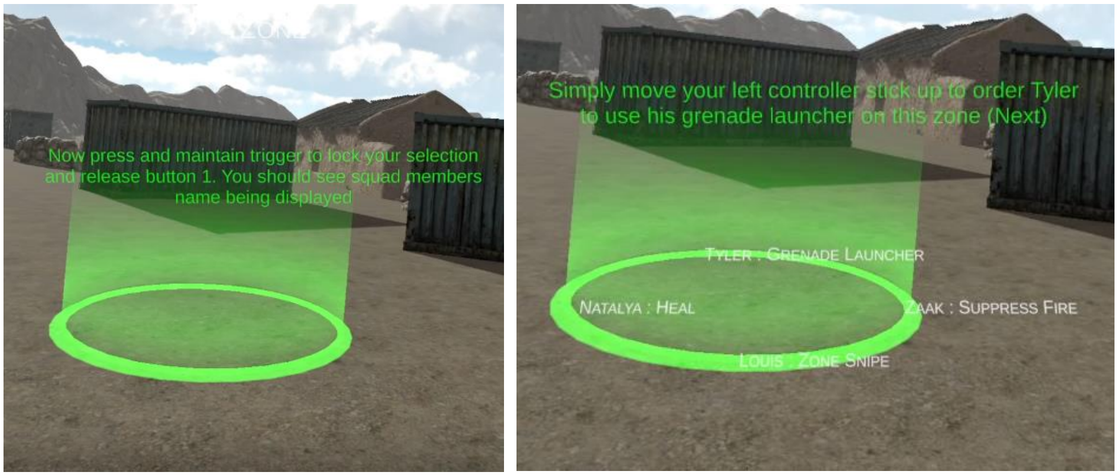
Another embodied map interaction technique is to directly pin the VR environment that the player is situated within. For example, as shown in Figure 167, when I point my index finger, which is simulated by holding the grip button while releasing the trigger button, a green circle appears in front of me on the ground. When I point to a possible cover, the circle turns yellow with a green arrow indicating a safe destination to move my squad.



Point your index on the ground 20 feets away. You should see a green circle on the ground (Next)

Figure 167. Point the index finger to a zone to select a possible cover in the game *Tales of Glory 2 – Retaliation*

In addition, the zone can also be used to order a teammate's special duty, such as grenade launcher, heal, suppress fire, and zone snipe. This design simulates the richness of the tactical orders that a commander can give to the crew. Still, it comes with the cost of the multiple-step text-based menu navigation controls with a combination of the joystick, multiple buttons, and the index-finger pointing control. In my experience, it took a considerable learning time to remember the menu operations and hierarchy. In the actual gameplay of assaulting an area, I only occasionally used the orders when I had to, such as have Natalya heal myself and ask for help when facing difficult situations of eliminating the enemies.



Now press and maintain trigger to lock your selection and release button 1. You should see squad members name being displayed

Simply move your left controller stick up to order Tyler to use his grenade launcher on this zone (Next)

Figure 168. point to the zone and assign a teammate's special job.

Compared with the "set waypoints" interaction in the same game, this zone interaction deals with the short-range and each member's job. They can compensate each other, although the interfaces and controls limit their efficiency.

Finally, in the *Population: ONE* example shown in Figure 169, the pinning action is simplified into two steps: selecting a pin icon in the pie menu and dropping the pin by aiming at the ground. This design improves efficiency by using the aiming cursor and does not give orders to a specific AI teammate. Instead, the pin is a shared vision among all player teammates in the squad. Despite not using the hand gesture or having the individual assignments, it still has a high simulation value with the pin metaphor.

Figure 169. Select a pin and aim to drop it on the ground in the gameplay of *Population: ONE*

6.5.4. Summary

As shown in Figure 170, the efficiency of these examples varies mainly on the time and steps of interaction. The hand-flick control and the tablet have relatively low efficiency because they potentially conflict with what their controls do for other interactions in the game context. As for the simulation aspects, the high-value examples have material references and realistic behavioral metaphors, such as the tablet, the pin, and the radio, while the low-value ones do not. These elements also reveal the tensions between the two values for this group of interactions.

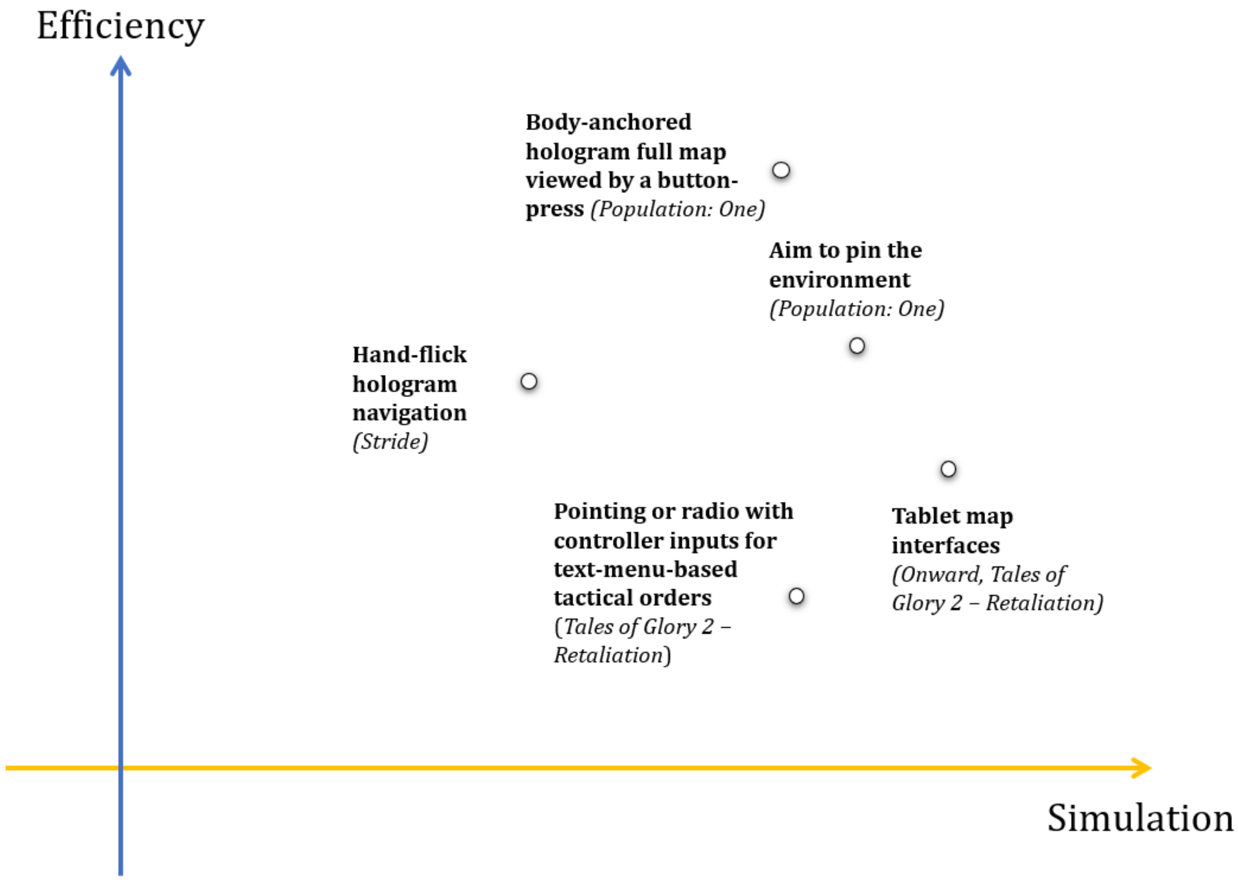


Figure 170. Map and signal interactions: summary and comparison based on the simulation and efficiency framework

7. RPG Combat Interactions

This chapter reviews combat interactions featured in the RPG genre, including weapon swing and abilities with melee weapons, parry and block, archery, spell cast, and use consumables. These interactions in VR all involve embodied gesture interactions and new techniques and mechanics that are different from key-press controls in traditional gaming platforms.

7.1. Weapon Swing and Abilities

Weapon swinging is one of the most prominent and defining combat interactions for VR games. Even out of the RPG genre, the hand-swinging action is successful in creating immersive and engaging gameplay. For example, the top-selling VR game – *Beat Saber* (Beat Games, 2018) features the core mechanic of swinging two glowing sticks to hit incoming blocks with the music and rhythm. In my opinion, the swing technique leads to an optimal result in the experience in that it achieves both high simulation and efficiency values. In VR action RPGs, weapons and abilities further embody the swing action and complexify the design space.

To comprehensively describe this design space and analyze the design values, I approach interaction examples selected from popular VR sword-fighting games from the following two perspectives: 1) swing mechanics, which is only based on the swing action but varies in direction, style, timing, and hit positions, and 2) special abilities, which deal with extended abilities with additional controls to co-function with swinging.

7.1.1. Swing Mechanics

In general, the speed and amplitude of swinging contribute to the damage it deals. VR games typically use intuitive wording such as “full swing”, “wider strikes”, and “heavy or light” attacks to inform players about this mechanic. The game *SWORDS of GARGANTUA*, for example, states that “the stronger you weapon you swing your weapon, the more damage you deal” and uses colored numbers to indicate the degree of the damage, as shown in Figure 171. Likewise, in *Until You Fall,* the wider you swing, the more damage you will deal. These two ways translate back to the speed and amplitude of the physical swinging and hold true in my experience with most VR sword-fighting interactions. There seems to be minimal speed and amplitude thresholds in the game system to distinguish an effective swing attack from a casual and weak movement.

Full Swing

The stronger you swing your weapon, the more damage you deal. The color indicates the degree of the damage (white < yellow < red)

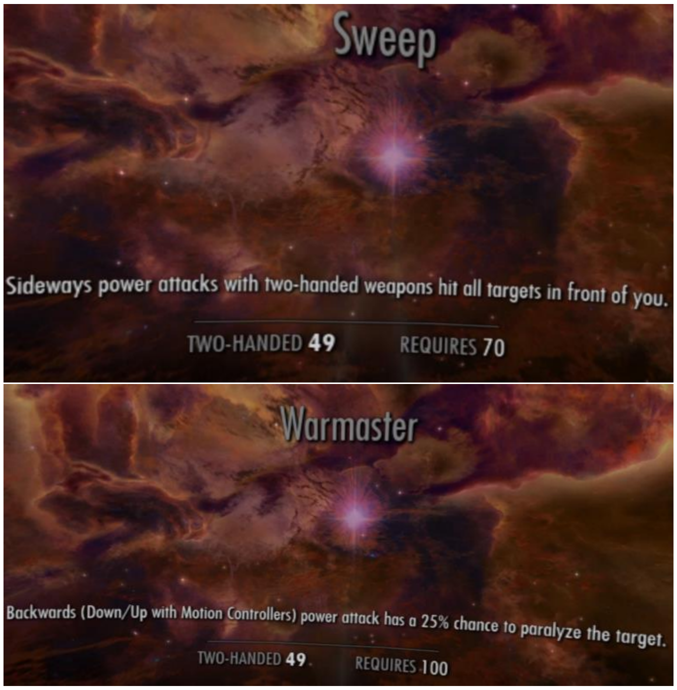
Swinging with Wider Strikes deals More Damage

Figure 171. Stronger or wider swings deal more damage in VR. Left image: the explanation of a full swing in *SWORDS of GARGANTUA*. Right image: swinging with wider strikes deals more damage in *Until You Fall*

The stronger or wider swing mechanic achieves high values of simulation and efficiency. And compared with more efficient button-press controls, it prioritizes the high interaction fidelity of hand movement in VR. However, it can get boring and tiring for players after repeating this physical activity for a long time. Therefore, the following variations and features come into play.

7.1.1.1. Direction

The direction of swinging can be an extra dimension in this design space. A representative example is the sweeping (sideways) and backwards (down -> up) weapon swings in *The Elder Scrolls V: Skyrim VR*. As shown in Figure 172, they are available as talents for players to unlock in the game to customize their playstyle and deal bonus damage. The sideways sweep talent allows hitting multiple enemies in front of the player and further improves the realistic simulation and gameplay efficiency. The paralyzing effect of the backwards is more of a game design choice than a realistic simulation, but it still leverages the backward direction as an alternative technique.



Sweep

Sideways power attacks with two-handed weapons hit all targets in front of you.

TWO-HANDED 49 REQUIRES 70

Warmaster

Backwards (Down/Up with Motion Controllers) power attack has a 25% chance to paralyze the target.

TWO-HANDED 49 REQUIRES 100

Figure 172. Two talents in *The Elder Scrolls V: Skyrim VR* for players to swing in sideways or backwards for special effects

This example in Figure 177 explicitly constrains the swing direction more than allowing it as a bonus variable. It simulates breaking the enemy's shield or armor to cause a critical strike. The visuals are necessary for this mechanic to work rather than be efficient. However, it reduces the degrees of freedom of swinging in any direction. The are irons rather than realistic weak points revealed by the enemy body itself.

CRITICAL STRIKES

Slash through Critical Strikes to Damage the Enemy

CRITICAL STRIKES

Slash through Critical Strikes to Damage the Enemy

Figure 173. Critical strikes and visuals in *Until You Fall*

7.1.1.2. Style

Style is another significant variable for the weapon swing interaction. It deals with options of *weapon types* and the dependent *wielding types* and *attack types*. The sweeping technique is an example that requires a two-handed weapon. In most VR sword fighting games, the two-handed weapons are more powerful than one-handed ones as the reward of using both hands, such as dealing bigger burst damage and breaking the target's defense. However, whether the player must physically grab a two-handed weapon with both hands or not is a design choice, and it reveals the tension between simulation and efficiency. As discussed in the introduction of this dissertation, *Skyrim VR* does not simulate the physical behavior of holding a two-handed weapon with two hands. As a benefit, it allows the other hand to cast magic and efficiently perform controller inputs in any relaxed position.

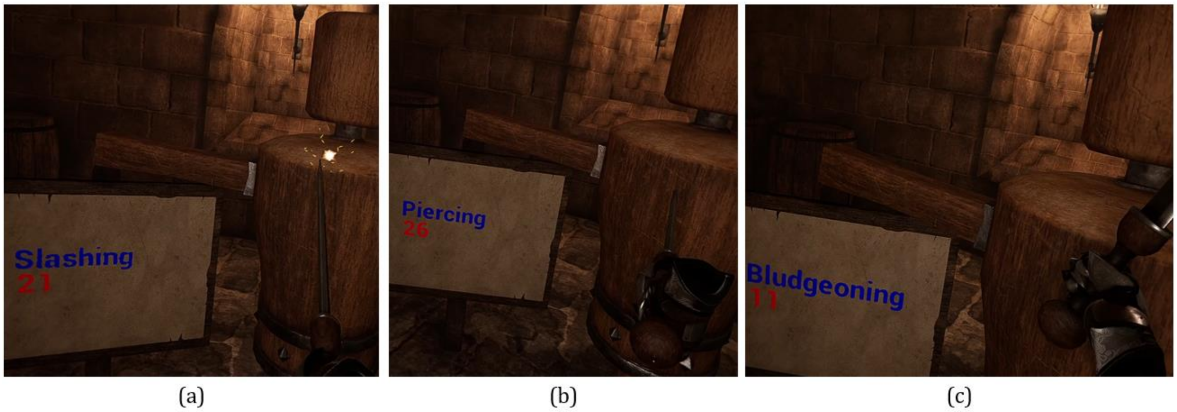


Figure 174. Three attack types in *Legendary Tales*: a) slashing, b) piercing, c) bludgeoning. In this example, the sword weapon is specialized in piercing, and hence piercing deals higher damage than slashing.

The game *Legendary Tales* implements a complex weapon swing attack system with multiple variables. First, the game has three melee attack types: 1) slashing, 2) piercing, and 3) bludgeoning, as shown in

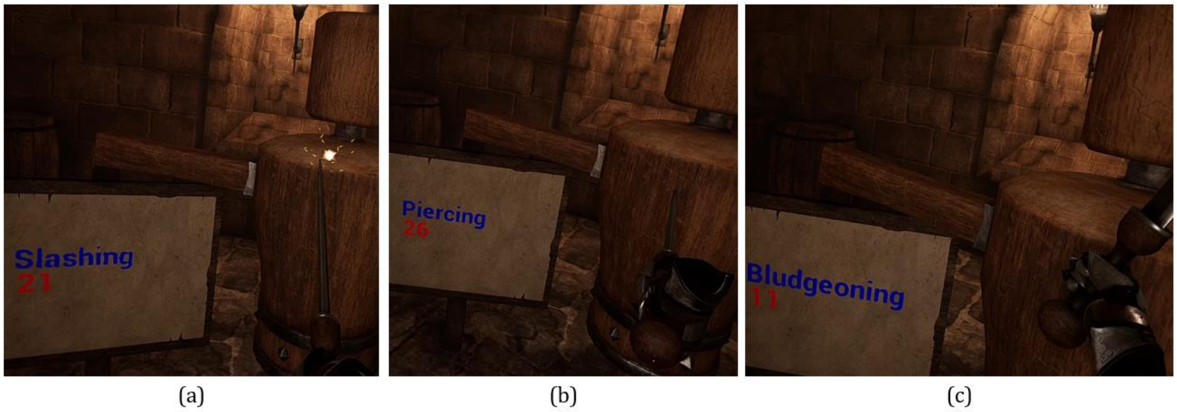


Figure 174. Each type uses a specific part of the weapon, slashing uses the blade, piercing uses the tip, and bludgeoning uses the rest blunt part. In addition, each weapon in the game can be specialized in one of the attack types to deal more damage than other attack types. In the Figure 174 example, the sword is specialized in piercing and deals 26 damage, which is higher than the 21 damage from slashing. However, piercing has a strict condition of having the sword stuck in the body, which is a good simulation despite being slightly more challenging to perform than slashing. The bludgeoning only deals 11 damage, which is also a good simulation of the poor swordsmanship.

Another variable that defines the style is weapon speed, which is determined by the weapon's weight. Light weapons swing faster than heavy weapons, and the faster the weapon is swinging, the higher the damage dealt. However, in my experience, heavy weapons can deal slightly more base damage per normal attack but are more difficult to swing. They need to be swung more slowly and can lead to a failed swing.

Combat skill is also a variable in the game's melee attack system. For example, with the "master of weapons" skill that comes with a sword gear (Figure 175), bludgeoning can stun an enemy, and slashing or piercing will render the next attack to be always heavy. Thus, this skill makes the bludgeoning no longer a "failed" low-damage attack and improves combat efficiency.

Master of weapons

When you do slashing or piercing attacks the next attack is always heavy and gives bonus damage equal to the player's wisdom and strength. When the enemy is bludgeoned they are stunned for a short time.

Figure 175. Combat skill "Master of weapons" as an additional variable in the weapon swing system of *Legendary Tales*

Finally, the game further balances the heavy vs light weapons and complexifies the weapon swing mechanics by introducing the cooldown time between each swing.

7.1.1.3. Cooldown and recharge

For example, the sword requires the player to wait 1.5 seconds between attacks; otherwise, it will end up with a "light attack" that only deals a small amount of damage. This mechanic prevents players from "cheating" the swing actions by repeatedly flicking their waist rather than performing wider swings that take longer between two consecutive attacks. Also, it allows light weapons to swing more frequently (in addition to swinging faster) to generate more damage per second (DPS).

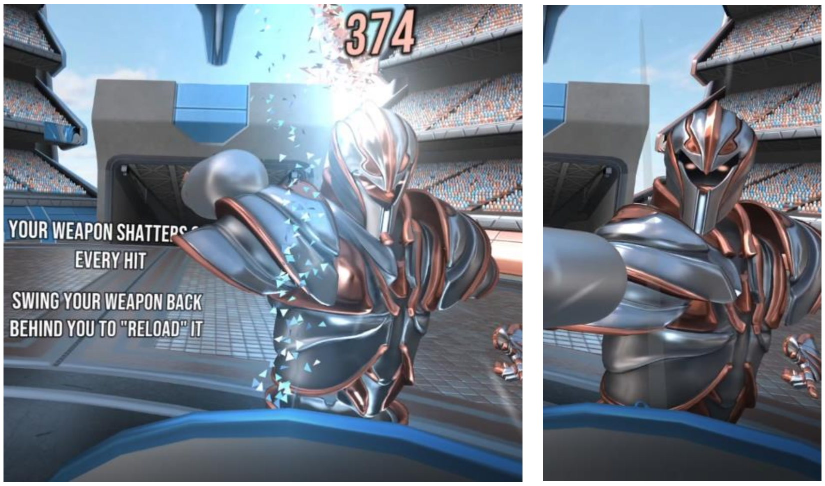


Figure 176. The reloading mechanic in *Ironlights*: the weapon shatters on every hit (left image) and will show an empty line visual if it is not reloaded (right image).

In a sword-fighting arena game *Ironlights*, the weapon cooldown is implemented as a “reloading” mechanic. As shown in Figure 176, the weapon shatters on every hit, requiring the player to swing it back behind their body to “reload” it. If the player does not reload, they will only see an empty line representing a non-existent weapon that will not hit anything. The shattered and semi-transparent line visuals are good simulations, with the digitally looking triangles being a commonly used metaphor of virtuality and shatters. However, the shattering and recharging effects are hyperreal in this particular game context. The “swinging back behind the body” mechanic is relatively easy to perform with inertia.

In addition, *Ironlights* has a time-slow effect during the swing to allow players to fine control their swings to hit the gap in the opponent’s defense. At the same time, it constrains the player from swinging too quickly (in addition to too frequently). As shown in Figure 177, the blade has a following visual of the actual physical swing. If players swing too quickly, the blade will not be able to keep up.

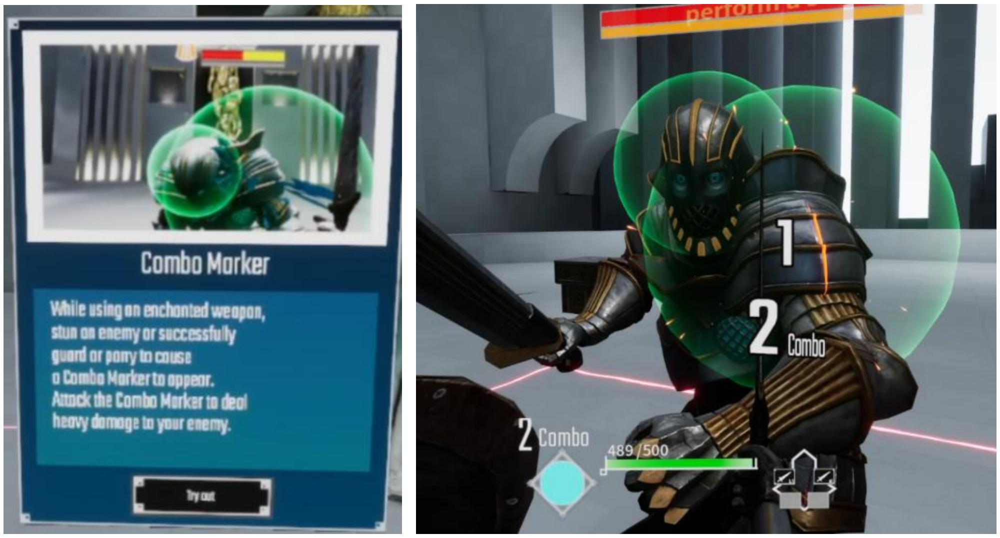
IF YOU SWING TOO QUICKLY, YOUR WEAPON WON'T BE ABLE TO KEEP UP

SMOOTH, SLOW, FLOWING MOVEMENTS WILL HELP YOU KEEP CONTROL

Figure 177. Time-slow effect in *Ironlights* to constrain the swing speed from being too quick but afford fine controls of its direction and position.

7.1.1.4. Hit positions: combo markers, weak points, and cut lines

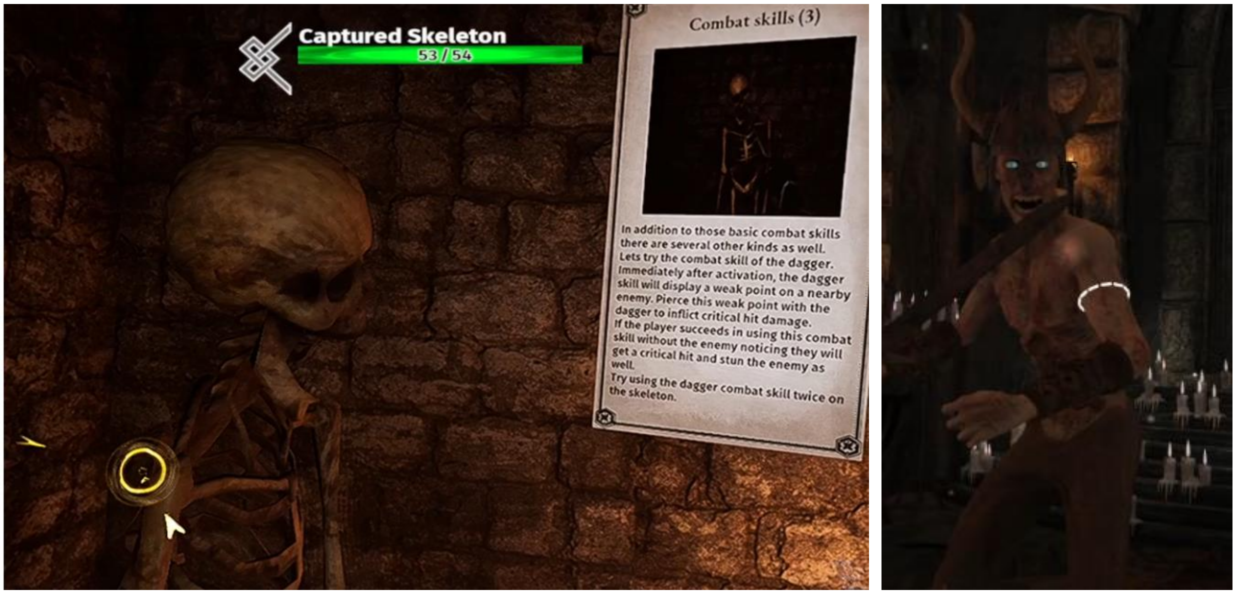
Finally, the precise hit position can be another design variable that enriches the embodied swinging interaction, and the visual interfaces on enemy bodies for assisting the first-person embodied interaction further reveal the value tensions. In the game interaction design, the hit position and its visual forms can be given different narrative meanings.



While using an enchanted weapon, stun an enemy or successfully guard or parry to cause a Combo Marker to appear. Attack the Combo Marker to deal heavy damage to your enemy.

Figure 178. Combo markers in *SWORDS of GARGANTUA*

In the example shown in Figure 178, the highlighted hit position translates to combo markers. In the game design, the combo markers appear after the player successfully performed other combat mechanics, such as stun and parry, and hitting the markers deals heavy damage. Although the mechanics make sense, and the combo makers simulate the combat dynamics involving the enemy's weakness and the player's "victory rush", the positions of these weak spots seem to be random without any animations of the enemy revealing these markers in some logical ways. In terms of efficiency, this example shows what I would like to call the "collider trick". The size of the effective hitting collider is larger than the corresponding part of the enemy body to improve the efficiency by trading the precision.



In addition to those basic combat skills there are several other kinds as well. Lets try the combat skill of the dagger. Immediately after activation, the dagger skill will display a weak point on a nearby enemy. Pierce this weak point with the dagger to inflict critical hit damage. If the player succeeds in using this combat skill without the enemy noticing they will get a critical hit and stun the enemy as well.
Try using the dagger combat skill twice on the skeleton.

Figure 179. Visual interfaces on enemy bodies. (left) an enemy's weak point in *Legendary Tales*; (right) a cut line in *Shadow Legend VR*

The weak point interface in *Legendary Tales* (Figure 179, left) has a similar visual interface but a different narrative. It is activated by a dagger's combat skill and can lead to special effects such as a stun and critical hit, which we will discuss soon. The visual form simulates a hole and informs the piercing attack type. The combat skill of the dagger also narrativizes the weak point's visual form to be something insightful observed from the ability and matches the idea of assassination. Compared with the combo markers, the interface visual is more precise (but the collider can still be larger). Although the weak points seem to appear on random positions on the enemy, the visual design reconciles the value tension. It achieves both high simulation and efficiency values.

In the right image in Figure 179, the enemy reveals a white cut line after being stuck in the combat some time, indicating the action of cutting its arm. This cut line can also appear on other limbs, and the neck will be the final part to finish the enemy[23]. Once the player successfully cuts through the line, the enemy limp will fall off and start bleeding. The cutting is a realistic simulation. The cut line is an efficient visual hint but does not look diegetic. The bloody and cruel content can be uncomfortable to watch for some players. Therefore, overall, the cut line design does not score in the highest ends of both values.

---

[23] For the purpose of not involving bloody, gore, and cruel graphics, I will not provide images for these examples but only briefly imply them using texts. To see the full content, readers can play the game or watch videos.

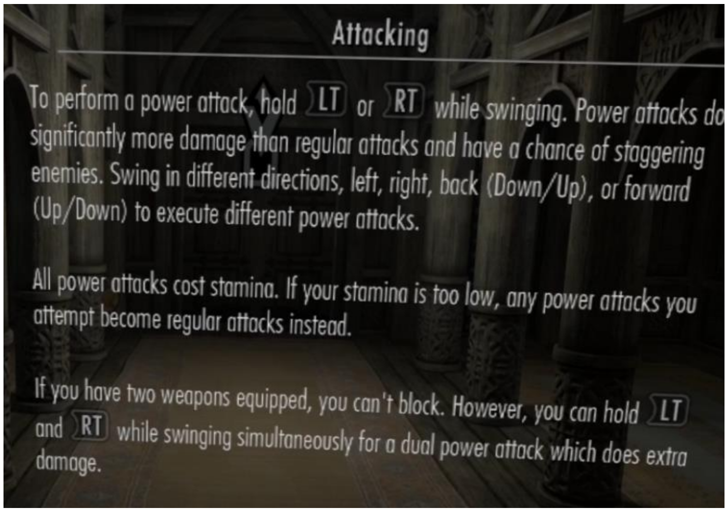
7.1.2. Special Abilities

This section deals with weapon swing abilities with *extra controls* in addition to the swing action. It includes melee abilities such as critical strikes, combat skills, enchanting weapons, and ranged abilities such as air blades and other super effects.

7.1.2.1. Button inputs while swinging for critical strikes and super effects

Pressing or holding a button while swinging can be the simplest way to add effects to a normal weapon swing attack. In *Skyrim VR*, holding the trigger button of the weapon's controller while swinging performs a power attack (critical strike), which deals significantly more damage at the cost of the player's stamina (Figure 180). Metaphorically, the control of holding the trigger button simulates the behavior of holding the weapon more firmly for a stronger attack. Combining the power attack with the dual-wielding style allows performing dual power attacks while losing the block ability for defense. Overall, this power attack multiplier achieves both high values in simulation and efficiency.

Besides the "holding the weapon more firmly" metaphor for the power attack, pressing the button can metaphorically connect to activating a super effect. For example, the weak point example from *Legendary Tales* (Figure 179) is the dagger weapon's combat skill, and it is used by pressing the X or A button.



Attacking

To perform a power attack, hold LT or RT while swinging. Power attacks d significantly more damage than regular attacks and have a chance of staggering enemies. Swing in different directions, left, right, back (Down/Up), or forward (Up/Down) to execute different power attacks.

All power attacks cost stamina. If your stamina is too low, any power attacks you attempt become regular attacks instead.

If you have two weapons equipped, you can't block. However, you can hold LT and RT while swinging simultaneously for a dual power attack which does extra damage.

Figure 180. Power attack mechanics in *Skyrim VR*

Another example of a trigger-based ability is activating the “super” effect, as shown in Figure 181. Although the trigger control does not simulate more than the concept of activating something, the visuals and the effect have carried the entire interaction design to achieve high values of simulation and efficiency. The color change in the weapon’s visual simulates an “enchanted” weapon with a super effect, informing the player that it has become more powerful than its normal state.

ACTIVATE LEFT SUPER

With your Weapon summoned, Press the Left Trigger to Activate your Left Hand's Super

(a) (b) (c)

Figure 181. The super effect of a dagger weapon in *Until You Fall.* (a) Activating by pressing the trigger; (b) color change in the weapon’s visual; (c) causing an immobility effect on the enemy after the swing hits it

Enchanting the weapon is a common mechanic in RPGs. It means infusing magic spells into the weapon to enhance it. The weapon’s visual appearance usually reflects the power. Besides the button-press control, VR has shown a more narrativized enchanting interaction by involving touch interactions with the other hand.

7.1.2.2. Interact with the other hand to enchant weapon

As shown in Figure 182, this enchanting interaction simulates the process of infusing magic power from the left hand into the sword using the special gauge resource accumulated in combat. Although there is no visual showing the gauge on the hand and its transmission into the weapon, the weapon visuals show an update during the enchanting. In terms of efficiency, the enchanting process takes about one second, and the actions with the two hands are relatively easy to perform. In addition, the left hand does not need to be empty as it looks in the tutorial.

Enchant

Using the Special Gauge, hold a weapon in front of your chest to Enchant it. Enchanting a weapon increases its attack power.

Enchant

Using the Special Gauge, hold a weapon in front of your chest to Enchant it. Enchanting a weapon increases its attack power.

Figure 182. In *SWORDS of GARGANTUA*, the player enchants a weapon by holding it in front of the chest and touching it with the other hand. The enchanted weapon does more damage and has special visual and audio effects.

In another enchanting example from *Blade and Sorcery*, the player can choose a specific type of magic spell and hold it on the weapon to perform the enchanting, as shown in Figure 183. This interaction design expands the use of magic in this game beyond throwing the spell to attack enemies. Compared with the previous enchanting example, this one more explicitly shows the magic power at hand. The direct interaction achieves high simulation and efficiency values.

Figure 183. Selecting a magic spell (left) and enchanting the weapon (right) in *Blade and Sorcery*

7.1.2.3. Ranged abilities and throwing attacks

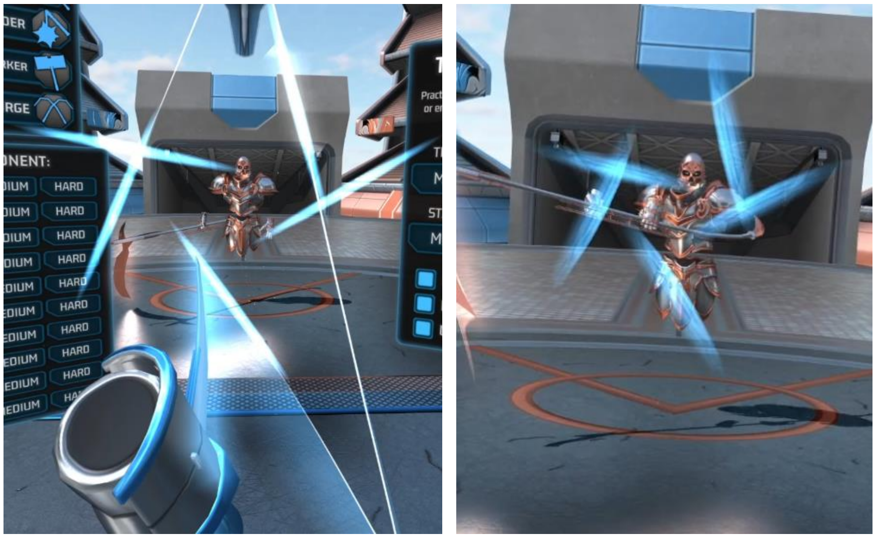


Figure 184. The air blade ability in *Ironlights*

Besides abilities that enhance melee attacking, the weapon swing can also cast various ranged abilities. For example, in *Ironlights*, players can swing the sword weapon to store “air blades” then release them to perform a ranged attack (Figure 184). Each air blade forms in the middle of each swing. In terms of simulation, the air blade concept and visuals are metaphorical. Still, the embodied controls of swinging and releasing the blades well match the concepts. In terms of efficiency, the gameplay is turn-based, and players can keep storing up to five or six air blades as long as they keep swinging the sword, and the swing-based mechanics are not difficult to perform.

Ironlights provides a wide range of weapon choices and demonstrates many ranged abilities. These ranged abilities do not simulate how the weapon works in the real world but still metaphorically connect to their affordances. The daggers, for example, turn into two handguns when activating their ranged ability, bringing efficiency in attacking the enemy (Figure 185).

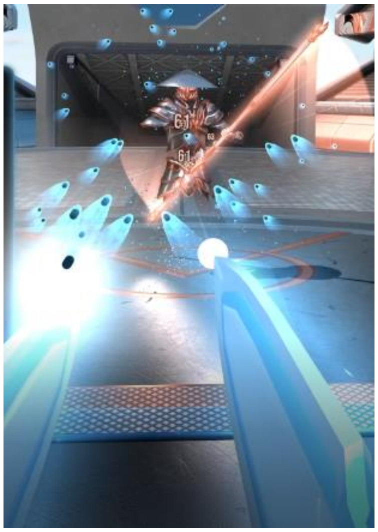


Figure 185. The ranged ability of the daggers in *Ironlights*

Besides the above examples, other games simulate a thrown weapon interaction by throwing a small weapon, such as a dagger or a knife. As we discussed in the grenade interaction, the throwing can be inefficient due to its physics simulation. The throwing trajectory can end up significantly different from how the player feels it should be. I experienced such inefficiency of throwing a knife in the game *Legendary Tales*. I failed most of the times when I tried to throw it on a target. It can be that my skill level is poor. But objectively, I find the throwing interaction lacks clear visibility of the directional vector in which the knife is being thrown when releasing the button.

The following example shows reconciliation to improve the throwing visibility by throwing the head of an axe during the swinging (Figure 186). The control feels easier because the object size is larger, and the long handle of the axe supports better visibility and physical simulation of throwing. As a result, the player has a better chance to preview the releasing point and estimate the direction. In addition, the axe's head returns to the handle automatically after it is thrown, and the automation further improves efficiency.

(a) (b) (c)

Figure 186. Throwing the head of an axe in *Shadow Legend VR*: (a) controls; (b) hitting a target; (c) the head part returns automatically. Screenshots from https://www.youtube.com/watch?v=V36DjjjduD8

7.2.3. Summary

In summary, the weapon swing interactions and abilities overall feature high levels of simulation and efficiency because they are directly based on the motion tracking affordance of VR controllers. Beyond the normal swinging, their variations are designed to be different attacking skills and abilities in the game context. These fun features reveal minor tradeoffs between the two values. The variations of the tradeoffs manifest in the direction, style, cooldown (timing), hit positions, and additional controller inputs.

I will not do a regular chart to put all the examples on and compare their nuance, with good reasons. First, most of them already feel optimal in both values and are fun to perform in the game. Second, there are too many of them, and they are diverse and very context-dependent. I have already grouped them by their similarity and controls in the above discussion and compared their values in detail.

7.2. “Pause!” - Consequences of Realistic RPG Combat Simulations

Here I want to pause our detailed examination for a bit and reflect on some of the broader consequences of the “optimal” combat interactions. These consequences are likely to have a long-term negative impact on the VR gaming culture and future generations of VR consumers. Therefore, I want to call for a pause before their exacerbation.

7.2.1. Intensifying Violent, Blood and Gore Content

Straight to the point, these realistic RPG simulations have the potential to intensify M+ content[24], especially the violent, blood and gore, cruel and anti-human graphics, and beyond them, the interactions to execute and produce them, in first-person and with your own hands, in an unparallel realistic fashion.

In searching for the examples for this section, I have seen many online gameplay videos of VR sword-fighting games featuring this kind of M+ content and experienced some on myself. The facts are:

1. Most of the creators of these videos, if not all of them, who commentate while playing the game, are male players.

2. These gameplay videos intentionally show various ways of torturing enemy bodies and showing dominance, such as cutting off limbs and heads and grabbing them, piercing their bodies and faces, burning and poisoning, and distorting bodies with hands or using tools.

3. They sound like enjoying these interactions as they try to make fun of them for the intended audience. The numbers of views and likes of these videos reach thousands and millions, making them top results of searching the game's name.

For example, when searching “blade and sorcery” on YouTube, the most viewed one titled “Blade And Sorcery VR Is An Absolute Nightmare - This Is Why” has 1.6M views, 59K

---

[24] M+ content in the gaming community means the game is rated in the “Mature 17+” or “Adults Only 18+” category that include discriptors such as violence and blood/gore, based on the standard rating categories and content discriptors set by Entertainment Software Rating Board (ESRB). https://www.esrb.org/ratings-guide/

likes and less than 800 dislikes at the time of writing this dissertation. The creator commented:

> "After writing this script I played this game non-stop for 3 days
> straight to get the footage
> I feel a bit...
> Different." (UpIsNotJump, 2019)

The top comments of this video highlight their favor, addiction, and "fun" aspects of the video. The rest top videos in the search result are like this one to a large extent. These videos successfully fooled YouTube's policy (*Violent or Graphic Content Policies - YouTube Help*, n.d.) by making the violent content seemingly fun and entertaining with enough "other" contexts.

Clearly, the most popular VR sword-fighting games are fostering and promoting masculine and violent values and culture. The realistic interactions are accountable for this phenomenon because of how they simulate body physics in detail. They raise the potential to create the content. Because of how realistic they look, it almost feels like watching real-life behaviors.

However, as painful as it may look for any unintended audience, when encountering such content in VR, the internal experience of the player can be mixed and fluctuating between sickness and numbness.

7.2.2. Mirror-Touch Synesthesia

Another consequence of realistic combat interaction when encountering violent and painful content relates to what is known as mirror-touch synesthesia. To put it simply, it means feeling similar pain in the same part of the body when seeing another person feel it ("Mirror-Touch Synesthesia," 2021). Although in neuroscience, it can be a rare condition to feel exactly the same level of sensation as another person, it is suggested to be linked with empathy (Banissy & Ward, 2007). A normal person can still feel at least some level of the sensation when seeing it closely, such as in a VR gameplay video or experience.

The synesthesia explains why it can be uncomfortable to watch the intense realistic violent interactions from VR. When it comes to encountering or experiencing them in VR, I argue that the mirror-touch synesthesia can be amplified by the medium featuring the senses of immersion and presence, the first-person perspective, and simulations of realistic interactions. The player can observe another person's pain, even if it is an innocent enemy, in a close range where the details cannot escape the eyes.

Furthermore, I would like to call it "the avatar synesthesia" when the "mirror" becomes the character that is (infinitely close to) the player in VR. When the player's character gets hit violently, the synesthesia immediately travels to the player's physical body because there is no distance to observe it from a third-person perspective.

The mirror-touch synesthesia, or the avatar synesthesia, highlights a source of uncomfortable experience in VR beyond the overly discussed motion-sickness, which many people took as a primary excuse for stepping into VR. As I mentioned, I am a player who is almost immune to motion sickness. However, in my experience, such as in the hit and die interaction examples and some sword-fighting content, I still feel uncomfortable, mostly from the mirror-touch and avatar synesthesia, which I further contend that they reveal the inefficiency design and the value tension with realistic simulation. I suggest that future design should avoid the potential for such synesthesia in VR and be cautious with realistic combat interactions. For this value tension, I do not think there are good ways to reconcile except to prioritize the comfort or the efficiency of comfortably engaging with the game over the realistic simulation.

7.2.3. The Evil First-Person

However, in some gameplay moments, players are forced to perform, react to, or accept the intense, realistic, violent interactions. The painful synesthesia can get overridden by ludic motivation. Players may rationalize their behavior by the magic circle[25], and I contend that VR as a medium has the unique power to foster "the evil first-person".

---

[25] In game studies and play theory, magic circle is a metaphor of the play space that suspends the rules in the real world and convinces players to follow the game rules (Huizinga, 1955).

For example, in my experience with the cut line interface on the enemy skeleton (see Figure 179 in section 7.1.1.4), I clearly remember that when I saw gameplay videos from others, I could feel the mirror-touch synesthesia was more than reluctant to face the blood and gore content. However, when I played that part myself, my experience fluctuated between being sick and attempting to ignore it before I submitted to the motivation to progress in the game and get rid of seeing the visuals by finishing the interaction. After that, I would tell myself, "*this is just how it goes in VR*", and become numb for the same interaction in the rest of the game. Also, there were cases where I had to react boldly without time to think about the actions. While VR makes the first-personal player vulnerable to violent and uncomfortable content, it at the same time can trigger their instinct of self-defense, even through seemingly evil behaviors.

My experience proves that the evil first-person effect. I still feel uncomfortable watching those gameplay videos, in which scenarios I know I have the choice to avoid. However, I can understand how other audiences may internalize those behaviors when watching them do those in the videos. Ien Ang provides several perspectives for understanding the relationships between the media and its mass audience, including the media effect of uses and gratifications and the reception analysis (Ang, 1995). The uses and gratifications approach assume that audiences are active in their choices of media content, and the reception analysis approach examines how audiences construct meanings out of the texts of the media. From this set of perspectives, future mass audiences of VR would *resist* these violent simulations rather than take them as a certain outcome of the media.

Still, I think realistic violence simulation is a negative property that designers should mitigate when designing VR games. Otherwise, it may have a long-term transformative impact on the consumers as VR becomes more and more prevalent in our society. It should be paused before VR is labeled as a realistic sin simulator if we want it to appeal to a larger public.

7.3. Defense

Defense interactions are another indispensable component in the combat action system for many VR RPG games. This section presents two categories of defense interactions – 1) parry and block, and 2) dodge - that I find revealing the value tensions.

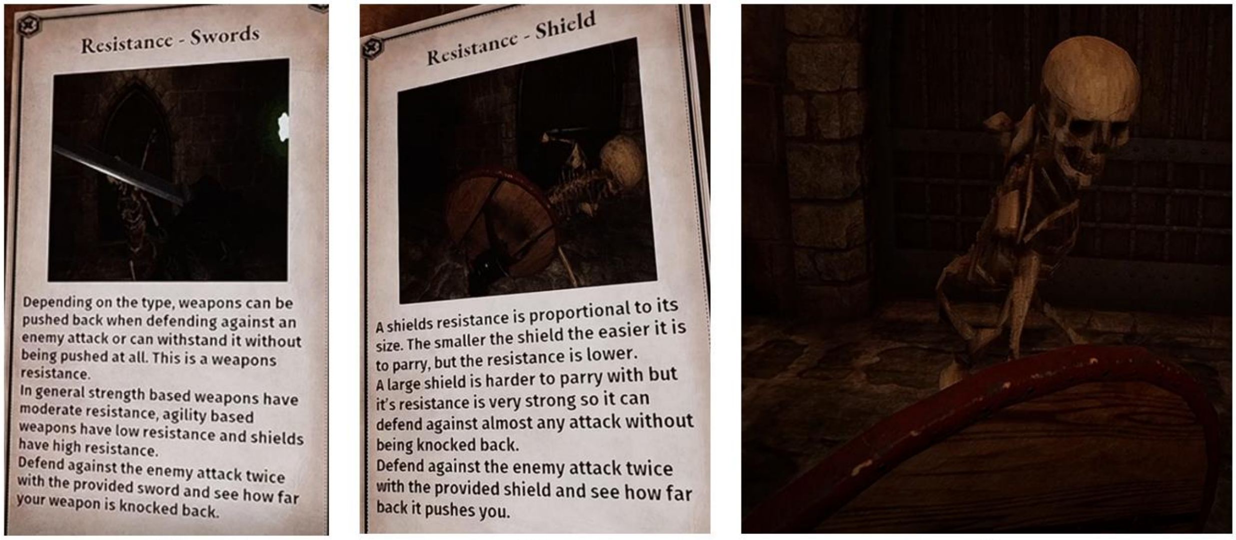
7.3.1. Parry and block

Parry and block are common defense interactions using either a weapon or a shield to counter an incoming attack. Some VR games use them interchangeably, and others implement nuance to separate them as two mechanics. Overall, I find shield block interactions, at the current state, are quite uniform - reaching out the hand holding the shield to block an enemy swing attack. It is a realistic simulation and is efficient, period.

7.3.1.1. Shield and size

Two exceptionally interesting design cases of shield blocking use size as the variable. The first case is shrinking the size of the shield every time it blocks an attack, as shown in Figure 187. Although obviously, it does not accurately simulate the real-world shield weapon, it presents the metaphor of a weakened shield that loses armor and makes it more challenging to precisely block incoming attacks.

Figure 187. Shield turns smaller on every block, making it more difficult to block incoming swings



Resistance - Swords

Depending on the type, weapons can be pushed back when defending against an enemy attack or can withstand it without being pushed at all. This is a weapons resistance.
In general strength based weapons have moderate resistance, agility based weapons have low resistance and shields have high resistance.
Defend against the enemy attack twice with the provided sword and see how far your weapon is knocked back.

(a)

Resistance - Shield

A shields resistance is proportional to its size. The smaller the shield the easier it is to parry, but the resistance is lower.
A large shield is harder to parry with but it's resistance is very strong so it can defend against almost any attack without being knocked back.
Defend against the enemy attack twice with the provided shield and see how far back it pushes you.

(b)

(c)

Figure 188. The resistance mechanic in *Legendary Tales* explains the advantage of a shield over a sword in defending and how the size of the shield further affects its parrying. (a) Swords will be knocked back when defending an attack due to their low resistance; (b) shields will almost not be knocked back due to their high resistance, the larger the shield is, the stronger its resistance will be but the harder to parry; (c) Parrying an attack with a small shield.

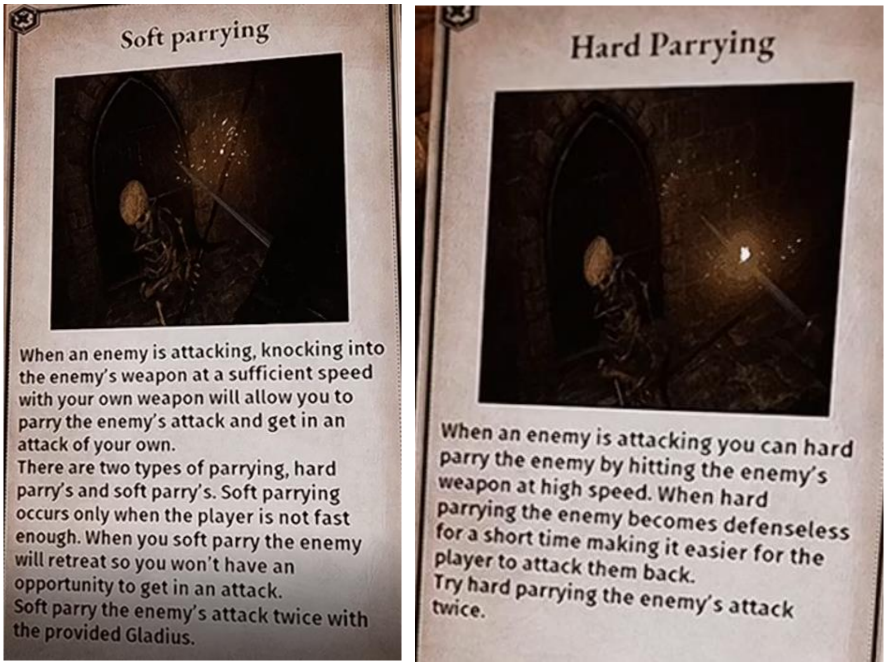
Another example of how size affects shield blocking involves the *resistance* mechanic in the game *Legendary Tales*. As detailed in Figure 188, resistance determines how far a weapon will be knocked back when defending against an enemy attack. In this case, "defending" means simply holding the weapon in front of the player to block an enemy's weapon swing. Shields have higher resistance than swords, making them more solid in parrying attacks. In addition, larger shields have stronger resistance but are harder to parry than smaller ones.

Compared with the size shrinking example, this design case brings in extra simulation value by reproducing the physical characteristics of weapons and shields. Doing so also achieves efficiency from resistance and makes both the small and large sizes useful in different scenarios.

Next, we will see more variations and diverse mechanics with parrying using a melee weapon.

7.3.1.2. Weapon parrying mechanics

Legendary Tales has four different types of parrying with a weapon. The first is simply holding the weapon in front of the player without moving it to block attacks, as mentioned in the resistance example. The three variations are 1) soft parrying, 2) hard parrying, and 3) half-swording.



Soft parrying

When an enemy is attacking, knocking into the enemy's weapon at a sufficient speed with your own weapon will allow you to parry the enemy's attack and get in an attack of your own.
There are two types of parrying, hard parry's and soft parry's. Soft parrying occurs only when the player is not fast enough. When you soft parry the enemy will retreat so you won't have an opportunity to get in an attack.
Soft parry the enemy's attack twice with the provided Gladius.

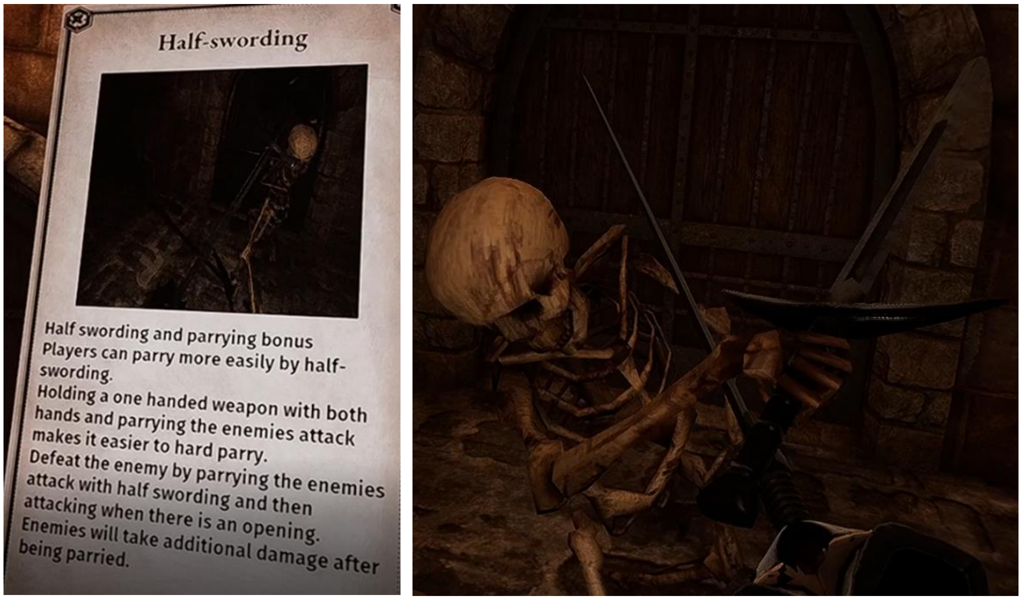
Hard Parrying

When an enemy is attacking you can hard parry the enemy by hitting the enemy's weapon at high speed. When hard parrying the enemy becomes defenseless for a short time making it easier for the player to attack them back.
Try hard parrying the enemy's attack twice.

Figure 189. Soft and hard parrying in *Legendary Tales*

As detailed in Figure 189, the control for parrying is to add a forward movement of the weapon at a “sufficient speed” to knock into the enemy’s weapon when it swings towards the player. Depending on the speed, it can trigger either soft parrying with a low speed or hard parrying with a fast speed. The advantage of hard parrying over soft parrying is that the enemy will be defenseless for a short time rather than retreat from the parry, making it easy for the player to attack them back.

Figure 190. Parrying an enemy's attack causes it to retreat



Half-swording

Half swording and parrying bonus
Players can parry more easily by half-swording.
Holding a one handed weapon with both hands and parrying the enemies attack makes it easier to hard parry.
Defeat the enemy by parrying the enemies attack with half swording and then attacking when there is an opening.
Enemies will take additional damage after being parried.

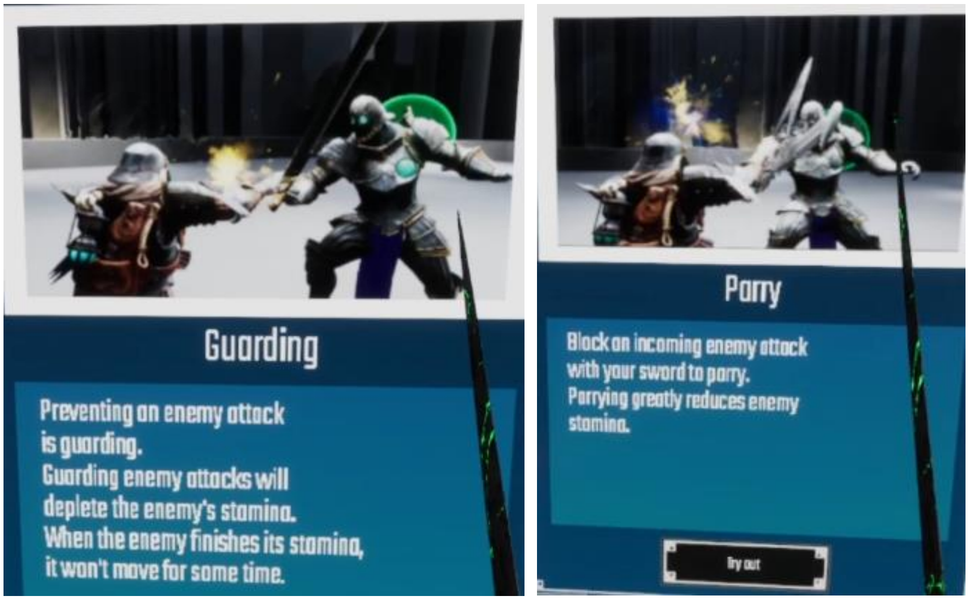
Figure 191. Half-swording in *Legendary Tales*: using both hands to hold a one-handed weapon makes it easier to hard parry.

In addition to the speed variation, half-swording is another technique using both hands to hold a one-handed weapon to parry (Figure 191). It makes hard parry easier than only using one hand.

Considering the basic blocking and three variations – soft parrying, hard parrying, and half-swording as a whole parrying system, I think it achieves both high simulation and efficiency values. In my experience, the fail rate of defending was low because at least I could block as long as I correctly predicted the direction of the enemy's attack. Any part along the blade is effective to parry and block, so given the length of the blade, players have a fair amount of time to react and position their weapon to defend successfully. The speed and two-hand control variations further create realistic physical simulations and reward players who perform extra efficient controls. The only case where a defense interaction can fail, as shown in Figure 192, is when the player fails to block the path of the enemy's swing towards the player.

Figure 192. A failed block where the enemy's attack bypasses the player's weapon

Comparatively, the parry interactions in *SWORDS of GARGANTUA* are less efficient because the passive defending action of holding the sword in front cannot block incoming swing attacks. To effectively defend with a sword weapon, players can guard or parry. As shown in Figure 193, the game does not explicitly tell the difference between these two in controls. They are similar to soft parrying and hard parrying, respectively. How their immobilizing effects work is sightly different from the *Legendary Tales* implementations. Guarding and parry reduce the enemy's stamina, and the immobilizing effect only happens when their stamina is out. This mechanic simulates the process of losing stamina instead of the dynamics in the hard parry, which has a more immediate binary effect for every parry.



Guarding

Preventing an enemy attack is guarding. Guarding enemy attacks will deplete the enemy's stamina. When the enemy finishes its stamina, it won't move for some time.

Parry

Block an incoming enemy attack with your sword to parry. Parrying greatly reduces enemy stamina.

Figure 193. Guarding and parry in *SWORDS of GARGANTUA*

Figure 194. Successful and failed guarding in *SWORDS of GARGANTUA*

In my experience with the parry and guard interactions, they often fail due to my slowness in reaching my arms out at a sufficient distance in front of me and moving it forward at a sufficient speed at the right timing. The timing can be confusing sometimes, and what I see is not what I get. As shown in the failed case in Figure 193, even I put my sword in front, the enemy sword still went past it[26]. Still, compared with the *Legendary Tales* examples, this shows relatively poorer simulation and efficiency.

---

[26] This can be either a bug, or just the mechanics of not moving the sword in the forward direction at the right timing. I did find with a shield or two swords, the parry went successfully more often than only having one sword, which seems to be weak

This last example from the game *Until You Fall* provides direct visual lines for players to align with to guarantee successful blocking, as shown in Figure 195. The line starts in blue color, with the white part filling from the two ends into the middle point, indicating the timing of the enemy's attack. If the player aligns one weapon with the line, the line will turn green, indicating a safe defense. These lines can appear one after another consecutively in a short time, or show up in a pair at different locations on the screen, to test the player's reaction speed of aligning their weapons.

Figure 195. *Until You Fall* provides visual lines for players to align with for guaranteed successful blocking

This visual design and mechanic solve the problem with timing and consistency. However, its efficiency sacrifices the simulation value of realistic parry and turns the interaction into a mini-game with those visuals, which can be boring after some time. I still prefer having the counter-swinging and the variations to reproduce the realness of the parry action.

---

against the two-hand weapon held by the enemy. However, the visual inconsistency was real, and on this point, I think its simulation is poor.

7.3.2. Dodge

Dodging is another common defensive interaction. To make a difference from a dashing locomotion interaction, the dodge interaction we will discuss in this section is performed in the context of quickly dodging an attack in combat.

Firstly, like the physical crunching, dodging can be performed physically by moving the head. As shown in Figure 196, the player has to dodge an incoming pillar to avoid it. Although only moving the head is sufficient to bypass it entirely, it may still seem to hit the player's body as it passes by. In this case, the head movement abstracts the full-body dodging, and this interaction design prioritizes efficiency over simulation. Especially when the player only tilts the head but keeps their body still, the value tension gets amplified by the visual-physical inconsistency in which the position of the player's body seems to be hit by the pillar, but in the game, it actually dodges the pillar.

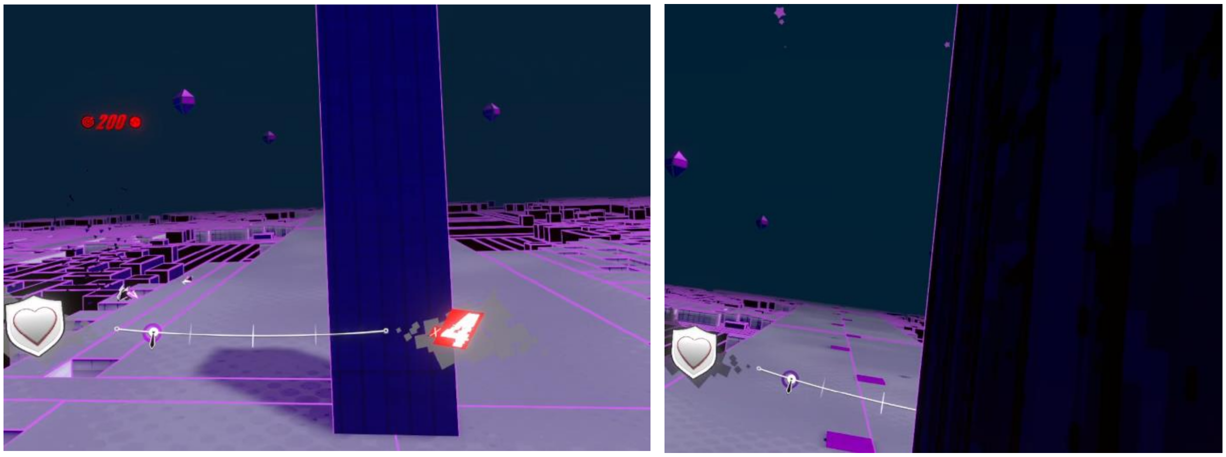
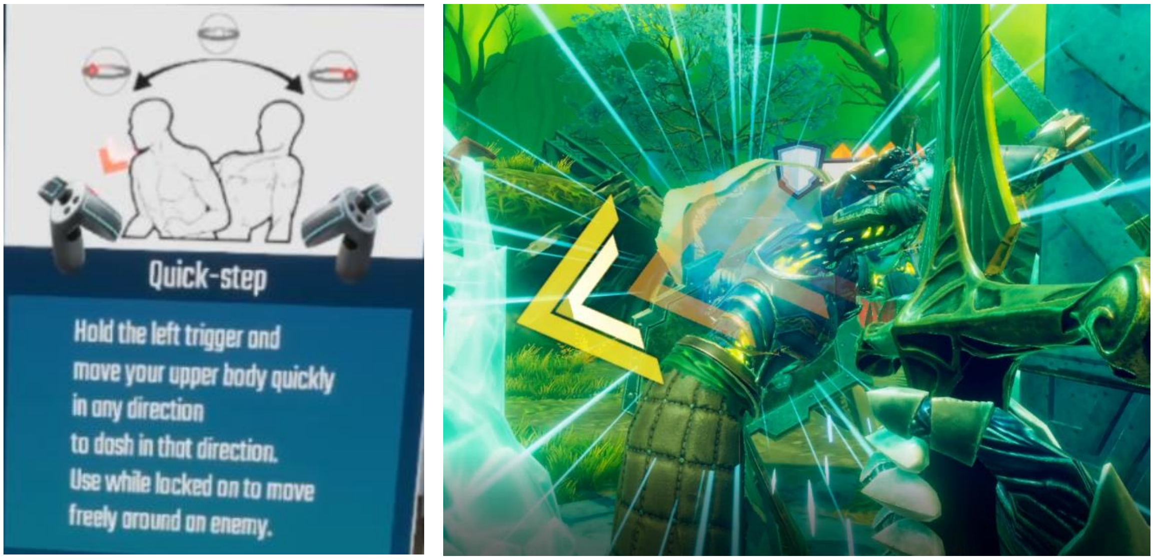


Figure 196. Physical dodge a pillar in *Pistol Whip* by moving the head

One way to reconcile the value tension is to use a lean-to-dash technique. As shown in Figure 197, the *SWORDS of GARGANTUA* implementation uses the left trigger button to trigger the state of quick-step, in which the player can dash in a direction by moving their upper body. This control is efficient because the head movement drives the dash locomotion of the entire body and is robust with the trigger button. In terms of simulation, the leaning upper body action matches the direction and the initial tendency of the dodge movement.



Quick-step

Hold the left trigger and move your upper body quickly in any direction to dash in that direction. Use while locked on to move freely around an enemy.

Figure 197. The lean-to-dash techniques in *SWORDS of GARGANTUA* (left) and *Until You Fall* (right)

Instead of using the trigger button to manage the dodging state, the lean-to-dash design in *Until You Fall* shows an arrow visual (Figure 197, right) every time a strong enemy attack comes to indicate the dodging time window and direction. Like the game's parry design, this design abstracts the sword-fighting combat into mini-games based on non-diegetic UI prompts.

7.3.3. Summary

As shown in Figure 198, the parry and block interactions score higher in simulation and efficiency values than the dodge interactions because the former is based on hand swinging and is further embodied by the weapons and shields. In contrast, the latter is based on the head or upper body's movement to drive the virtual character's body to dodge.

Within the parry and block interactions, the *Legendary Tales* example scores the highest values with the realistic simulations of block resistance of weapons and shields and variations of parrying mechanics, including the swing speed for soft and hard parry, the two-hand half-swording, and the immobilizing effect on the enemy after a successful hard parry. The align-with-lines interaction in *Until You Fall* is highly efficient in the gameplay to simplify the timing for the player. Still, it loses simulation value for using the non-diegetic visual prompts. The *SWORDS of GARGANTUA* example has relatively low efficiency because it does not have simple blocking and, in my experience, has a high failure rate in defending the enemy attack[27].

Within the dodge interaction group, the lean-to-dash techniques are more optimal than the physical head-only dodging because the latter may suffer the physical-visual inconsistency problem where only the head is dodging, but the character's body feels still in the position under the attack. The visual prompts again cause a tradeoff to prioritize efficiency over simulation, which makes the gameplay like an overly guided mini-game. The trigger control is the noticeable implementation for managing the state machine of the head's leaning action. Still, it may have limited application in other games' control schemes where the trigger is assigned to perform other actions.

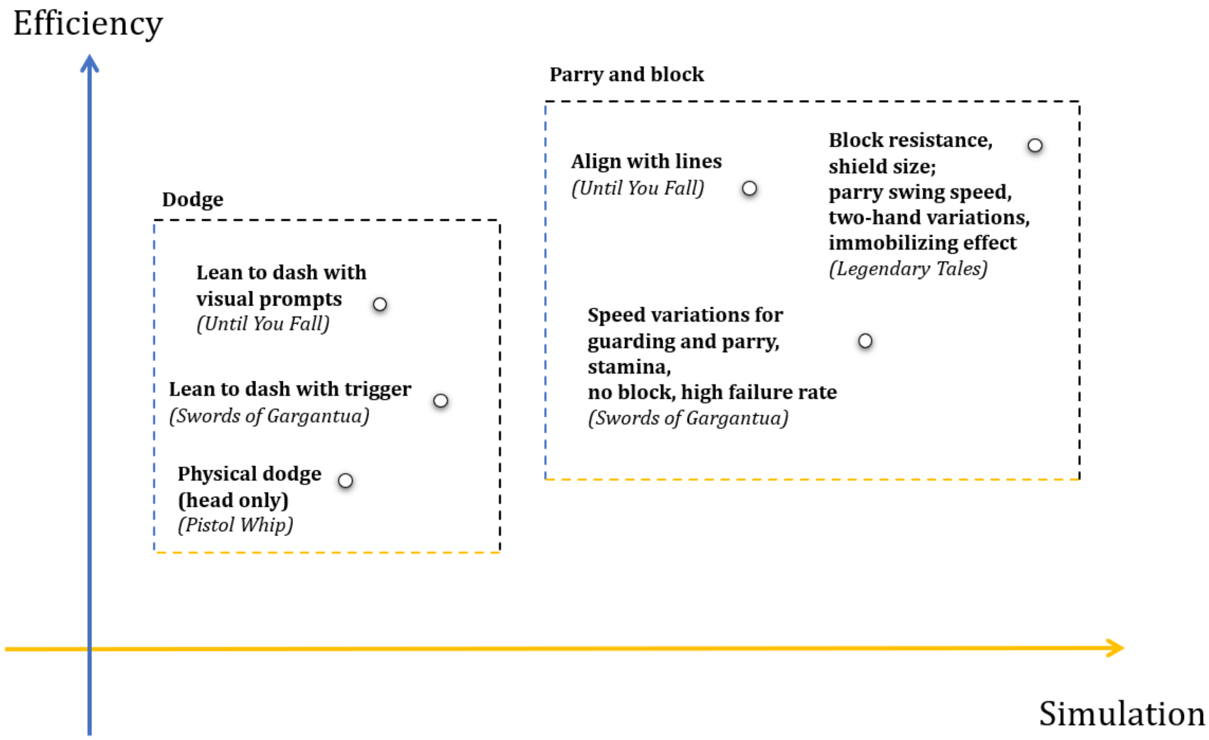


Figure 198. Defense interactions: summary and comparison using the simulation and efficiency framework

---

[27] A possible cause that worth further investigation is whether this is due to enemies using two-handed weapons in *Swords of Gargantua*. The parry and block examples in *Legendary Tales* and *Until You Fail* are mostly in the case where enemies attack with one-handed weapon.

7.4. Archery

Moving from melee attack and defense interactions, in this section, I will focus on archery as a ranged attack interaction using a bow or a crossbow.

7.4.1. Examples

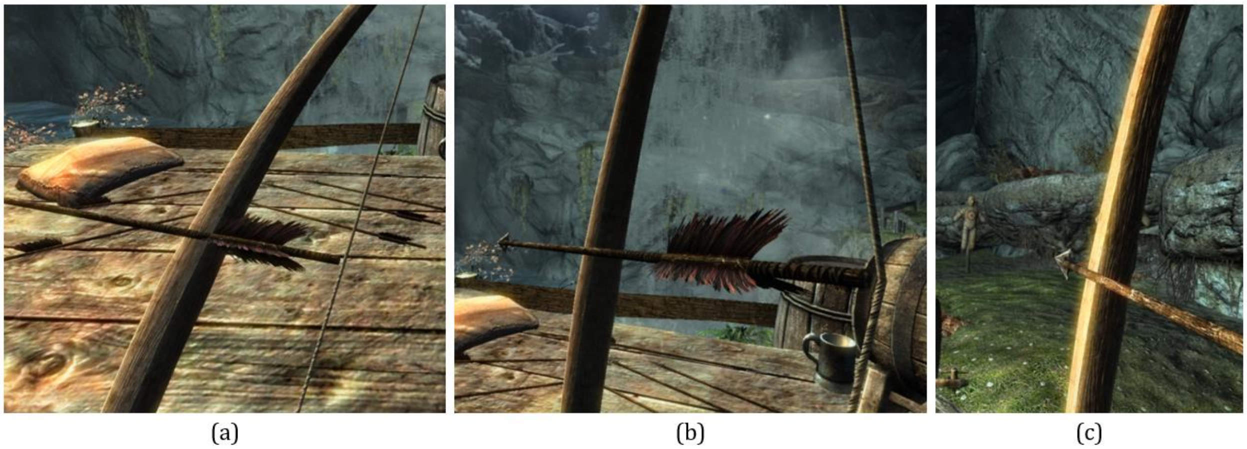


Figure 199. Archery in *Skyrim VR* and steps of shotting an arrow with the bow: (a) Bring the arrow to the bowstring, then press the hold the right trigger to nock the arrow; (b) draw the bow back; (c) aim at the target before releasing to fire.

Figure 199 shows the basic steps of archery interactions, including nocking the arrow, drawing the bow back, and aiming to shoot. These interactions have achieved high simulation values in terms of behavioral mimicking and interaction fidelity. The nocking and pulling actions are done by holding the trigger button, which maps to the finger actions. The nocking also uses the snapping mechanic to increase efficiency.

However, the aiming efficiency of the bow is low in my observations and experience, especially when compared with the gun aiming interactions. There is no visual aid or cursor for aiming, and the bow and arrow are flexible in both vertical and horizontal directions. For a target at a far distance, players also have to estimate the gravity effect on the arrow and its traveling speed, which depends on how far the arrow is pulled on the bow.

The "aiming shot" design on the bow shown in Figure 200 improves realistic simulation and aiming efficiency. The design makes it easy to hold the bow strictly in the middle of the screen and align the arrow's forward direction with the target without the bow blocking the sight.


Figure 200. Examples of aiming slot design for the bow: (left) *Legendary Tales* and (right) *Shadow Legend VR*

The *Budget Cuts 2* archery features the hyper-reality technology-themed bow and the arrow, as shown in Figure 201. The arrow is a sharp crystal that needs to be nocked in front of the bow. The drawing back action does not pull a material string or the crystal. Instead, it is a hyper-realistic laser beam that simulates the bowstring.

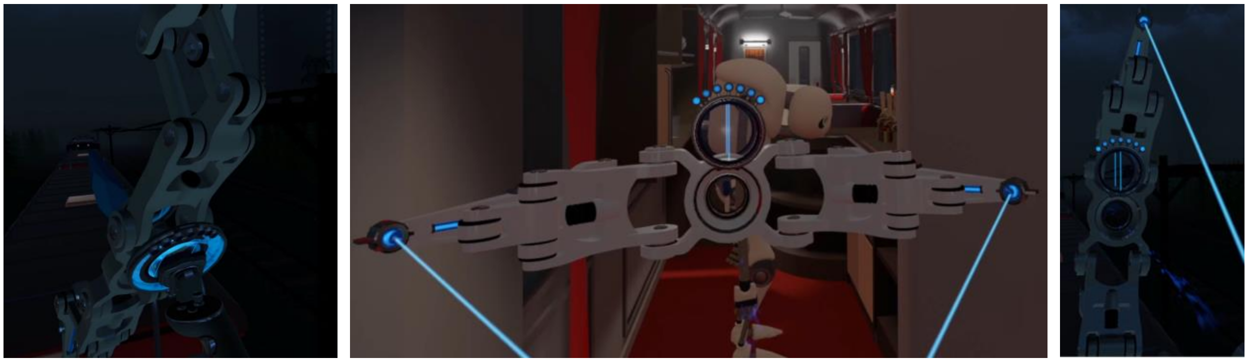


Figure 201. The hyper-realistic bow and arrow in *Budget Cuts 2* with visual aids for aiming.

The aiming lens of the bow is what improves the aiming and shooting efficiency. The blue dots light up as the string is pulled to the maximum, indicating its strength and the speed of the crystal. The lens displays additional laser beams that indicate the bow's direction and help players align with the target. Furthermore, the lens rotates with the bow and always remains on top of the “arrow” to facilitate aiming.

Another hyper-realistic archery that improves the efficiency of hitting the target is the multi-arrow design in *Sairento VR*, as shown in Figure 202. The three arrows increase the chance of hitting the enemy on one shot and can be used at any angle when the player rotates the bow. While it does not simulate the single arrow, I feel it keeps the same level of simulation value given that multi-shot is a legit ability in video games.

Figure 202. *Sairento VR*’s archery with multiple hyper-realistic arrows on the bow

The game context is another dimension in the design space of VR interaction, and the following examples design gameplay challenges that suit the mechanics of archery.

Figure 203. A gameplay challenge that suits the mechanics of archery in *Shadow Legend VR*

For example, as shown in Figure 203, the challenge is to kill the skeleton who keeps throwing bombs periodically at the player. In this case, the skeleton itself is not easy to kill by arrows or weapons, but at the same time, it does not move. Therefore, this is a perfect ranged attack scenario where shooting the bombs stacked at its feet to explode them is the easiest way to kill it and proceed in the game. Although players can miss the bombs several times, they only need to keep trying while avoiding the bombs thrown at them. The further the player stands, the longer it takes the bomb to reach, so they have a longer time to aim.

Figure 204. Unrealistic arrow trajectory and auto-aiming in *I Expect You To Die 2*

Another example from *I Expect You To Die 2* implements an aiming assist mechanic for its crossbow in the gameplay challenge of killing remote enemies (Figure 204). The game is set in an operation stage at a theater, and is played in the seated mode. The player ruins the show and has to escape safely by killing enemies that shoot arrows and throw poison grenades from a distance. Unlike the above examples in the RPG setting, where players have plenty of arrows, in this case, the player even has to use the shield to "borrow" arrows from the enemies. Therefore, the aiming assist can be rationalized as reducing the difficulty of the challenge. In addition, the crossbow automatically draws the arrow back as the arrow snaps to it and only requires one hand to pull the trigger to fire. These efficiency-oriented design features keep the player in the flow state of progressing through the game content rather than being skilled at the archery and aiming mechanics. The efficiency values also tend to make me ignore the unrealistic simulation of the arrow trajectory with the aiming assist.

Figure 205. Using the shield to borrow arrows from enemies in *I Expect You To Die 2*

7.4.2. Summary

As shown in the framework in Figure 206, I use the *Skyrim VR* implementation as a baseline archery example. I think it achieves a moderate/high level of simulation but relatively low efficiency. In my experience with it, I missed most shots because of no aiming aid whatsoever. The aiming slot adds simulation and efficiency values to the baseline implementation, but the bow's physical and realistic nature can still make it challenging to hit targets.

The examples that achieve higher efficiency value make archery easier to hit targets. The aiming lens on the hyper-realistic technology-themed bow *in Budget Cuts 2* and the multiple arrow example in *Sairento VR* both improve efficiency and further simulate additional themes and functions. The *I Expect You To Die 2* example has several efficiency-oriented features that fit the unique gameplay setting and challenge with a scarcity of the arrows, but its simulation value can be negatively affected by the aiming assist, if one cares about the unrealistic arrow trajectory.

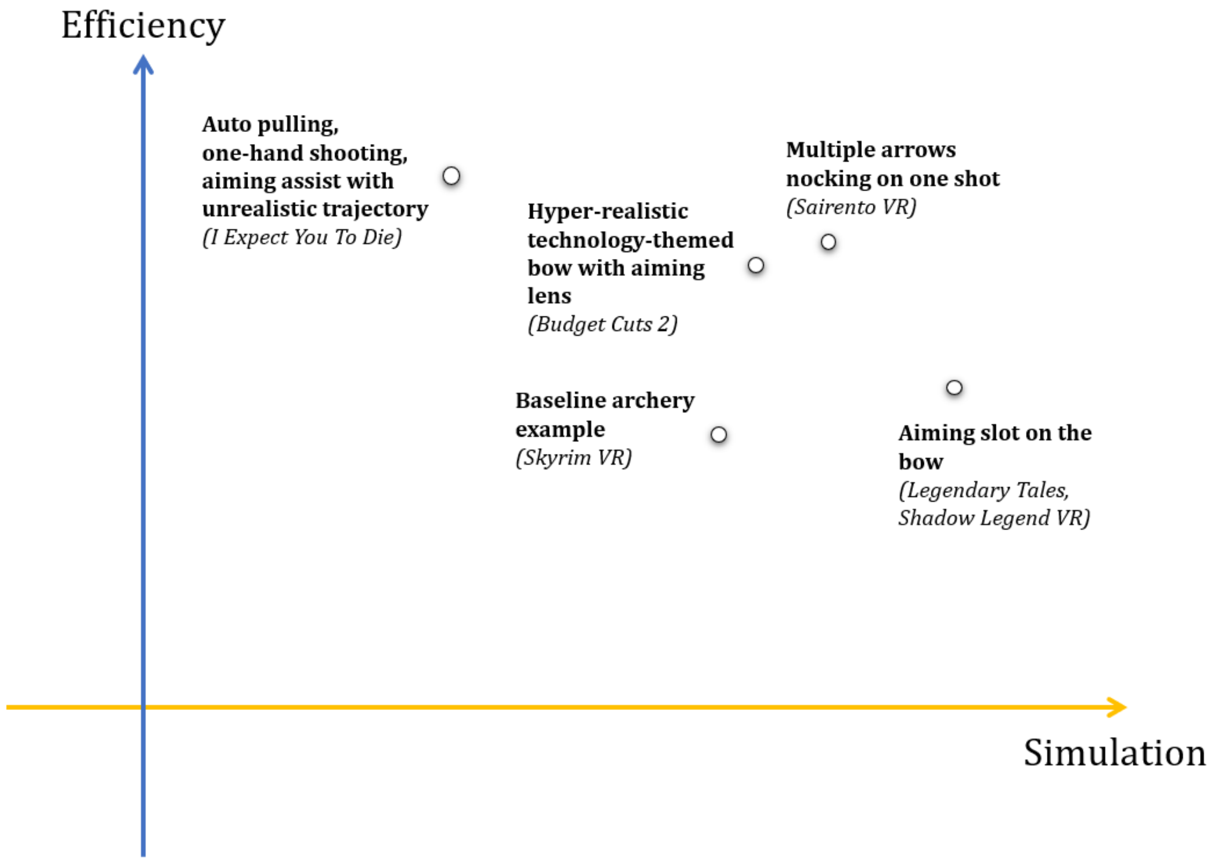


Figure 206. Summary and comparison of archery interactions using the simulation and efficiency framework

7.5. Magic and Spell Cast

Besides archery, spell casting is another typical ranged attack using magic and elemental power. This interaction typically can be broken down into three phases: 1) the selection phase and 2) the casting phase, and 3) the shooting phase. In the selection phase, the player selects a type of magic power to “equip” on the hand, typically from a selection menu. In the casting phase, the player casts the spell to build up its power in their hand. Finally, in the firing phase, the player throws the spell in a direction controlled by the hand movement and releases it to fire, and if possible, further controls the spell’s movement with gestures.

7.5.1. Examples

In *Blade and Sorcery*, players hold the X or A button to open the spell selection menu and touch a spell to select and equip it with the respective hand, as shown in Figure 207. The magic selection wheel will keep open at the hand when the player holds the button. Upon selection, the menu vanishes with the magic particle effects, with the selected spell remaining active on the hand.

Figure 207. Selecting a spell from the symbolic menu interface in *Blade and Sorcery*

During the casting phase, players hold the trigger button to cast the spell. As shown in Figure 208, the longer the player casts, the stronger the spell appears.

Figure 208. Casting spells with the hand: the lightning example and the fireball example in *Blade and Sorcery*

Finally, when the casting is finished and the spell power is built up, players can release the hand and throw the magic in any direction to fire it, as shown in Figure 209.

Figure 209. Firing the spell power when throwing and releasing it from the hand: the lightning example and the fireball example in *Blade and Sorcery*

This example is used as the baseline to compare with the following examples. Overall, it embodies the three phases of selecting, casting, and firing and achieves high simulation and efficiency values. The simulation is not mimicking real-world behavior that does not exist but an imagined spell casting technique established in movies and video games.

Furthermore, the use of magic can be extended in many ways. For example, both hands can cast magic, and some more powerful spells can be unlocked by combining the two spells from both hands. Figure 210 shows an anti-gravity field effect that can levigate the objects and the enemy when players combine the same void spell from both hands.

Figure 210. The anti-gravity field by combing void spells from both hands in *Blade and Sorcery*, screenshots from https://www.youtube.com/watch?v=z3JmikF52K4

As we discussed before, magic can be used for enchanting weapons, and this can be done by a single hand casting the spell while holding the weapon in *Blade and Sorcery*. The complex combinational magic mechanics give rise to many violent combat interactions, as shown in Figure 211. The player can further control the levigated enemy with gestures and enchant the weapon with the fire spell to attack the enemy. Also, the enchanted staff gains the new ability to hit the ground to create a power wave that knocks back nearby enemies.

Figure 211. Complex combinational magic mechanics, such as levigating magic and flame-enchanted weapons (left), give rise to violent combat interactions in *Blady and Sorcery*. The enchanted staff can hit the ground to knock back the nearby enemy (right). Screenshots from https://www.youtube.com/watch?v=z3JmikF52K4

Legendary Tales also implements fusion magic from both hands. Its magic selection interface "Magic Sigil" and the interaction of combining two spells by moving two hands close together are similar to those in *Blade and Sorcery*, as shown in Figure 212.

Figure 212. Selecting the fire spell and creating a fusion fire magic in *Legendary Tales*

The rune tablets are how players acquire new magic spells. They also display the game metrics of the magic and, most importantly, the gestures to fire them (Figure 213).

FireArrow

Damage	0 ▸ 10
Mana	0 ▸ 10
Level	0 ▸ 1

Condition

1 INT

FireBall

Damage	0 ▸ 10
Mana	0 ▸ 40
Level	0 ▸ 1

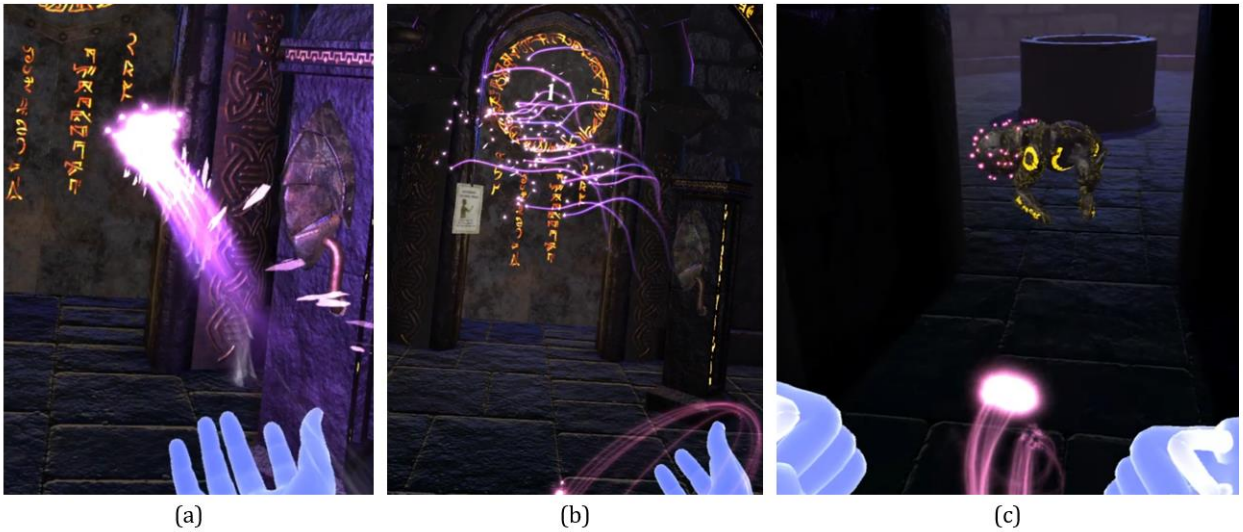
Condition

1 INT

Figure 213. Tune tablets that the play can acquire to learn FireArrow and FireBall spells and their gesture controls in *Legendary Tales*

The gesture control is designed to be a variable for firing different attack effects of the same spell. However, in my experience, the throwing has low efficiency in precisely controlling the direction of the fire arrow or the fireball. This is the general problem with throwing anything in VR, and I repeat it here to show that magic spells are no exception. In the video that I took screenshots for the *Blade and Sorcery* examples, the fireballs seem to have an automatic tracking mechanic to hit the enemies or a direction control mechanic by the hand when flying[28].

In the game *Waltz of the Wizard: Natural Magic*, the natural magic particles can be further controlled with hand gestures to allow changing their directions after the player releases them, as shown in Figure 214



.

Once they hit an object, they conduct it, allowing the player to remote control the object's motion with gestures. This extra control extends the firing phase and improves the simulation and efficiency, especially efficiency, which is reflected mainly by how easily the magic ball can hit the target in all these game examples.

---

[28] The video commentary mentioned that the gameplay used third-party mods, and I did not fully explore the original combat gameplay as much as I wanted to because of its violent content, so I left with a bit of ambiguity in this mechanic and only interpreted the design based what I saw in the video (https://www.youtube.com/watch?v=z3JmikF52K4).

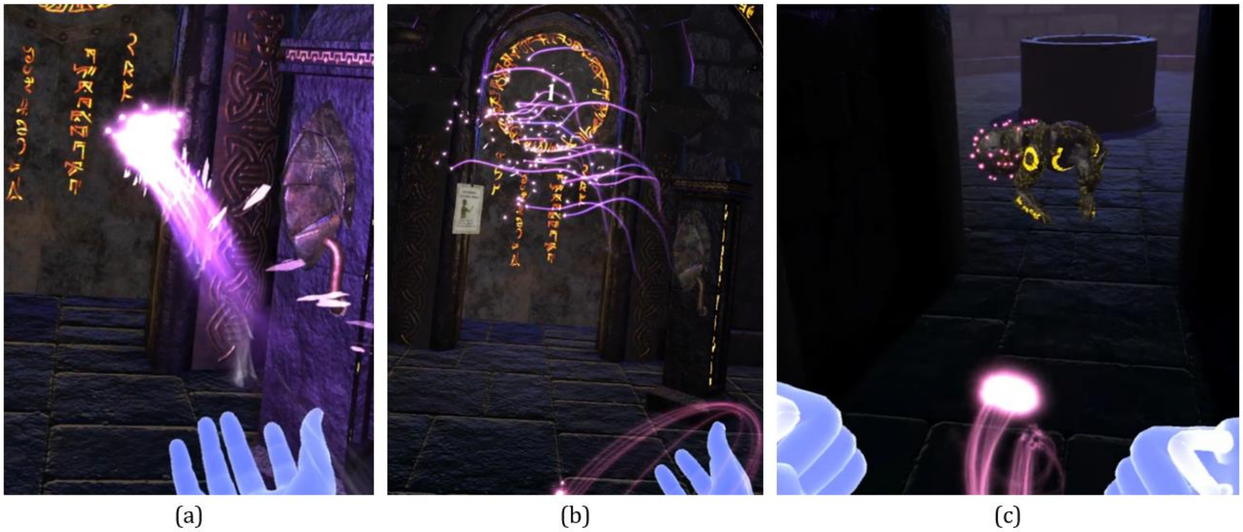

(a) (b) (c)

Figure 214. Magic interactions and hand controls in Waltz of the Wizard: Natural Magic: (a) releasing a cluster of magic particles, (b) guiding their further movement and directions with hand gestures, (c) conducting an enemy and controlling its movement. Screenshot (c) is from https://www.youtube.com/watch?v=qkvz3D3Ivy4

In addition, the game has two unique magic interactions shown in Figure 215. First, it matches the context well to gather natural magic particles from the air by closing the fist and pulling quickly with the palm facing the head. The game only has one magic power – the natural magic. Therefore, there is no need to summon a selection wheel. The interaction of the pulling power from the air simulates the gathering action and reinforces the natural theme. However, the gathering requires a fast pulling speed and can fail if it is not fast enough, and therefore is less robust compared with the selection wheel interface. Still, this hand-based movement does not significantly impact the efficiency value, and I feel its simulation value has a higher impact on the experience.

The second unique interaction is the sonic scream. Based on the conducting mechanic of the natural magic, this interaction further extends the design space by enchanting the mouth of the embodied player. It is unique because the microphone input is used to create a reverb effect of the player's own voice. The scream can also interact with objects in the game, such as ring a bell, but is not used as a primary combat interaction.

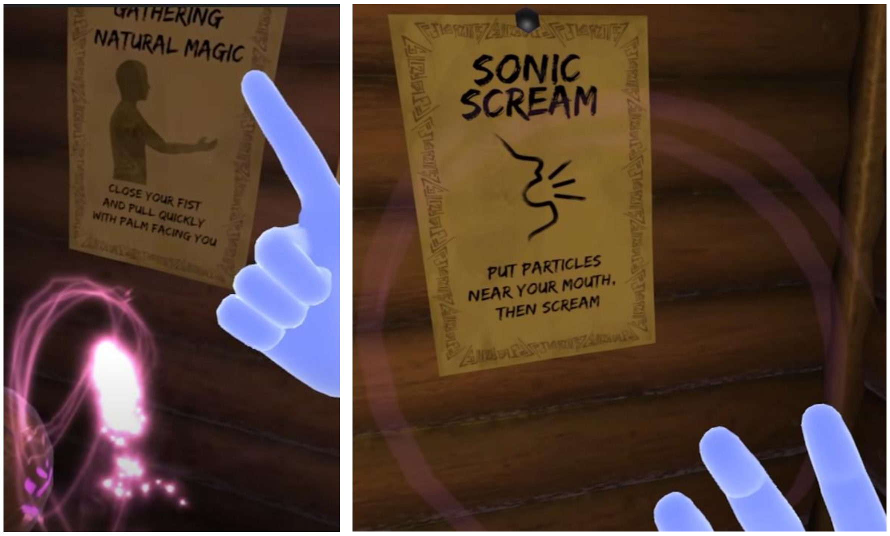


Figure 215. Two unique magic interaction techniques in *Waltz of the Wizard: Natural Magic*: (left) gathering natural magic by pulling quickly from the air; (right) sonic scream interaction using the microphone input after "eating" the natural power by putting the particles near the mouth. Screenshots from https://www.youtube.com/watch?v=9rDDhbaKE_8

In *Skyrim VR*, to use magic, the player has to navigate in the menu or the favorite list to select a scroll (Figure 214). Opening the menu pauses the game, and from this perspective, it is efficient in saving time in combat. However, the menu navigation is not the most efficient and does not simulate any magical theme like those selection wheel interfaces.

However, each scroll can only be used once. So, after I cast the fire balls but still did not do enough damage to kill the enemy, I found that the scroll of fire ball was no longer in my inventory and could use the scroll of blizzard as the spell casting option (Figure 217). The one-time-only use of scrolls makes magic and spell casting an expensive resource in the game and further limits its efficiency value.

A fiery explosion for 50 points of damage in a 15 foot radius. Targets on fire take extra damage.

Figure 216. Multi-step menu navigation to select a scroll before casting its magic spell in *The Elder Scrolls V: Skyrim VR*.

Targets take 20 points of frost damage for 10 seconds, plus Stamina damage.

Figure 217. After using the Scroll of Fire Ball, it disappears in the inventory, meaning that the scroll is for one-time use only.

The text-based menu navigation follows the design pattern in traditional video games, but it may not be the most optimal magic selection interaction in VR. It reveals the value tension of sacrificing the simulation value. The value tension is amplified by the need to manage a large number of items, including the scroll items themselves that are for using magic. Instead of adding a new magic selection interface, which may not apply to the scroll-based magic system, the game strives for consistency for all item selection interactions by using the same menu hierarchy. The value tension also gets amplified by the scarcity of the controls in this open-world RPG. The buttons on the controllers, which are used in previous examples to open an exclusive magic selection wheel, are taken by other actions. However, one benefit of the menu interface for selecting magic is the detailed description of the spell effect and its game metrics.

7.5.2. Summary

In summary, the magic and spell casting interaction examples have overall achieved high simulation and efficiency values with gesture-based controls. As shown in Figure 218, the outliner is the *Skyrim VR*'s magic interaction, whose scrolls can only be used once and need to be selected by pausing the game to navigate the menu.

Among the top-right region in the framework, I set the *Blade and Sorcery* example as the baseline. The magic system in this game is one of the most comprehensive and well-received ones in the community. The spells are overall easy to perform and hit targets, which is the essential efficiency criteria for this interaction. The magic system is also highly scalable and extendable. The spells can be used to enchant weapons, and some spells can be mixed and combined to create more powerful effects.

The other two examples have extra gesture controls and variations, such as casting different abilities with different gestures and further navigating the particles after releasing them. For these, I rank them higher in the simulation dimension. However, in my experience, I find they require extra skills to perform and are not easy to master to hit the targets. Therefore, their efficiency value is lowered due to the simulation of gesture interactions.

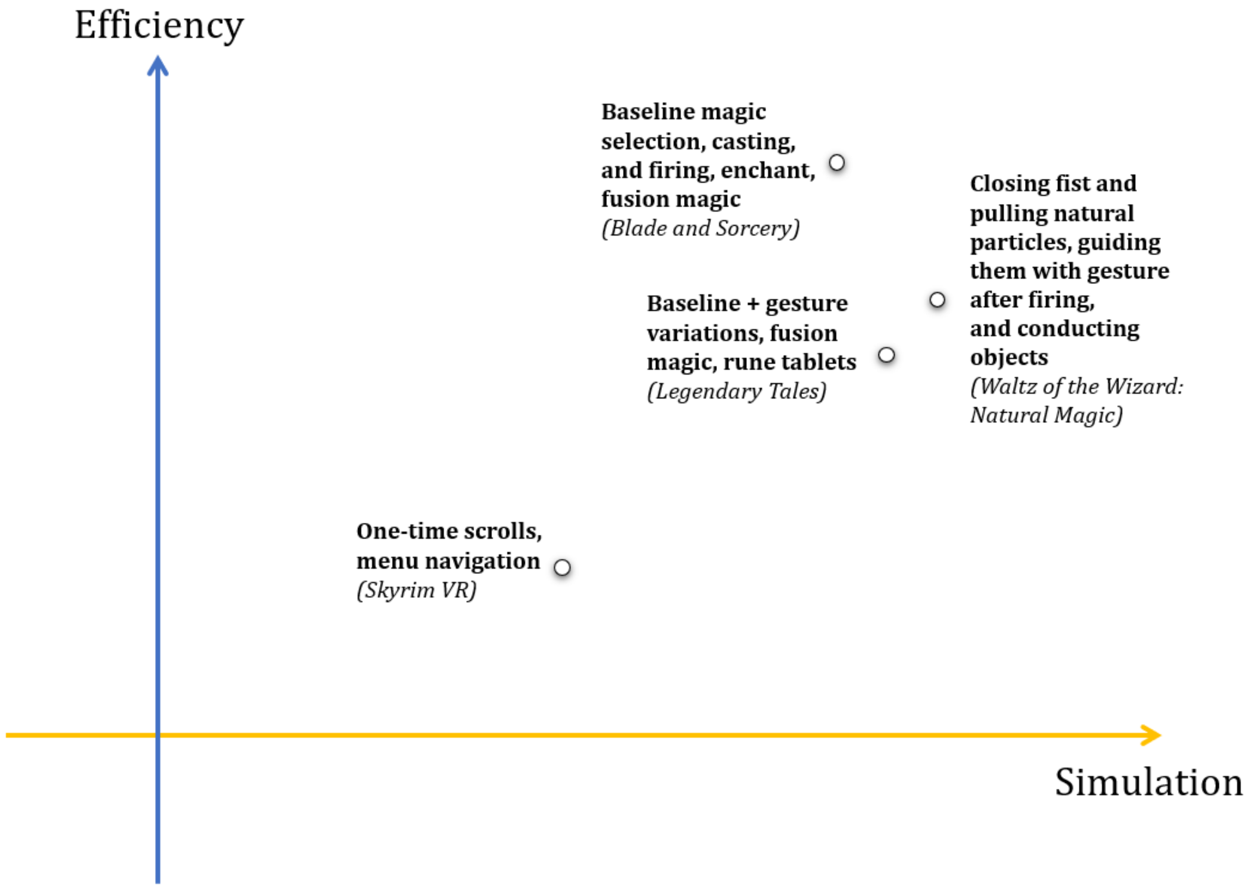


Figure 218. Summary and comparison of magic and spell casting interaction examples using the simulation and efficiency framework

7.6. Consume and Recover

Using consumables to recover health, mana, stamina, etc., is another everyday interaction that players perform in RPGs. This section will examine how these interactions balance the simulation and efficiency values in their adaption to VR.

The following subsections list three ways of consuming and recovering. First, the "eat and drink" approach is based on the interaction of grabbing a consumable food or drink and moving it close to the player's mouth. The second approach is to interact with recovery orbs floating in the game scene. The third features using a tool, such as an injector and a pair of defibs, to heal oneself or rescue a teammate.

7.6.1. Eat and Drink

In the introduction chapter of this dissertation, we mentioned the banana and soda examples in *Population: ONE*. I briefly revisit them as a starting point for the discussion. The banana requires players to peel all four slides before it can be "eaten" by the mouth (Figure 219), while the soda is more straightforward with only the opening interaction for drinking it.

Figure 219. Peel all four sides of a banana and bring it to the head to eat in the game *Population: ONE*

As a direct comparison, the eating banana interaction in *Hellsplit: Arena* skips the peeling phase, as shown in Figure 220. However, with the above peeling experience, I felt unrealistic when eating it and seeing its peels were left open on the ground (it automatically dropped on the ground after eating). Thus, the two examples reveal the value tensions in simulating the object's detailed structures and the abstraction of the meaningful action. The peeling example has higher simulation but lower efficiency, and the non-peeling example is precisely the opposite.

Figure 220. Eating a banana without peeling the four sides in *Hellsplit: Arena*, but its peels can be seen left on the ground after eating.

Nevertheless, I feel the control of eating the consumable has already done a good job of achieving *realism*. In fact, the direct (non-peeling) eating approach is more commonly implemented with other uniformly shaped consumables, such as an apple and a cake. The feedback also adds to the simulation and efficiency values. For example, when the player puts a cake close to the mouth in *I Expect You To Die 2*, there is a crispy sound effect with the visuals of some small pieces popping out, as shown in Figure 221, indicating that the cake has been consumed.

Figure 221. Eating a cake and seeing visual feedback of some small pieces popping out in *I Expect You To Die 2*

Furthermore, the physical shape of the food affects how it is designed to be eaten in VR. The carrot example in Figure 222, for example, supports this pattern, as its long shape affords the animation and the states of being consumed multiple times.

Figure 222. The long-shaped carrot is designed to be able to eat for multiple times in *Shadow Legend VR*

Similar concepts of abstraction and structural details are reflected in drinking as well. In the soda example in *Population: ONE*, players drink the soda in one shot by simply opening it then bringing it to the head, as shown in Figure 223.

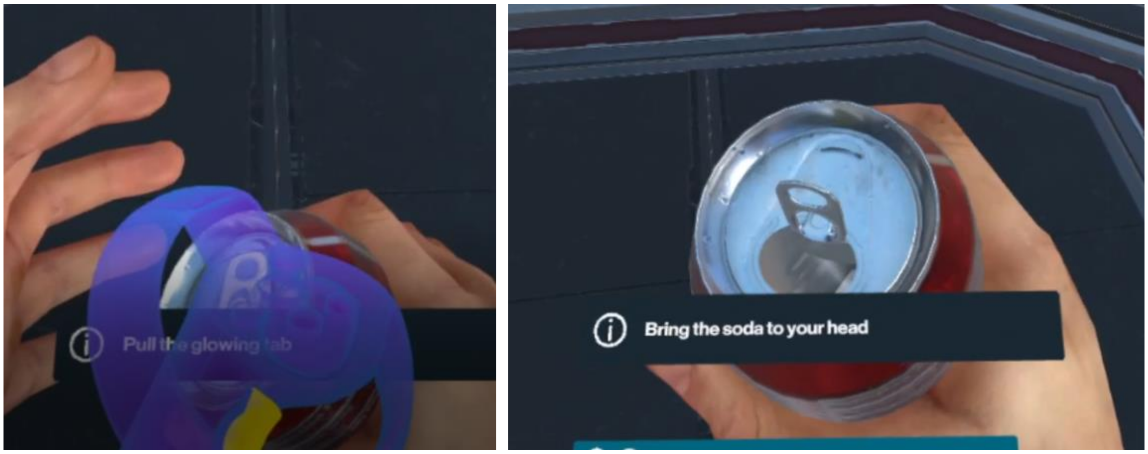


Figure 223. Drinking a soda in *Population: ONE* by pulling the tab and bringing the soda to the head

As a comparison, other drink interactions feature a continuous process of drinking. For example, as shown in Figure 224, drinking is performed by raising the water bottle high over the mouth and pouring the water down. The pouring is a continuous process, with the visual feedback of water and drops.

Figure 224. Drinking interactions in *I Expect You To Die* 2: pouring water from the bottle (left), with water drop seen near the mouth (right)

As shown in Figure 225. in *Hellsplit: Arena*, players can equip health potions in their waist slots or grab them at spots with floating air to support grabbing in the scene. To drink the potion, they still need to raise the potions high above the head to use gravity to pour the potion into their mouth. However, all the liquid in the bottle is consumed quickly in a one-time fashion rather than a continuous process.

Figure 225, Health potions in *Hellsplit: Arena*

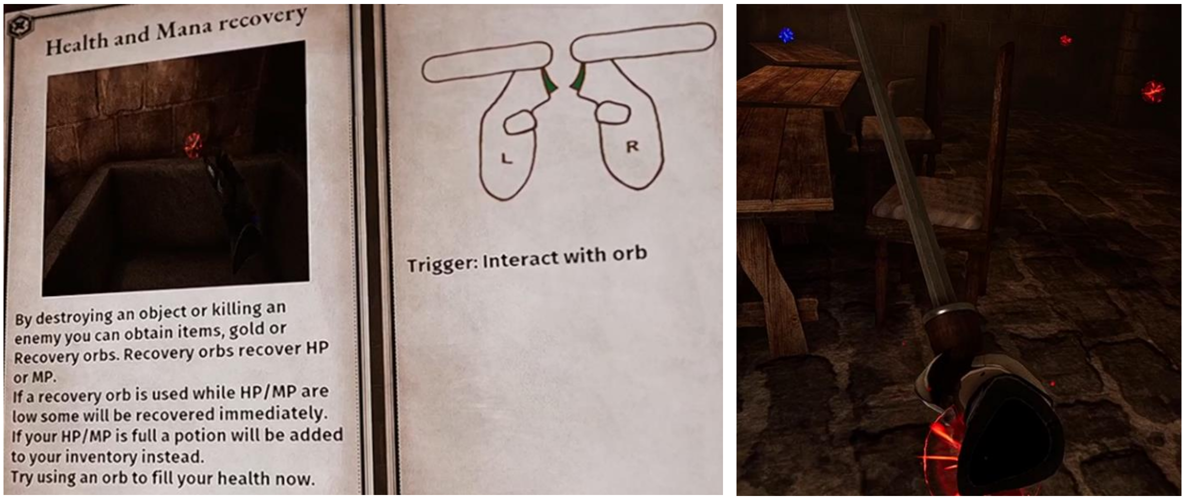
In terms of simulation of the drinking interactions, two platform constraints are worth pointing out. First, the platform does not know exactly the height of the mouth in the physical world. It can, at its best, estimate the mouth position based on the center of the headset. Second, it may be technically challenging to simulate the fluid property of the liquid. Therefore, it can be difficult to perfectly mimic continuous drinking by holding the bottle slightly above the mouth to pour the liquid. In my option, the interaction of raising the potion higher above the head to trigger the drinking overly simulates the pouring and is inefficient.

Finally, as a contrasting implementation, the food and drink can be consumed through the inventory menu interaction, as shown in *Skyrim VR* in Figure 226. This implementation is due to many items available to carry with the character in the game, and it skips the embodied interaction of pouring the liquid. However, the menu interaction can still be inefficient; it also breaks the immersion and misses the chance to simulate the realistic behavior.

Figure 226. Using consumables through the inventory menu

7.6.2. Recovery Orbs

Interacting with recovery orbs dropped in the scene is another design pattern in traditional RPGs, and it is typically done by moving the character onto the orb. In VR games, this interaction translates to touching the orb directly, usually with the hand, and pressing a button to use, if necessary.



Health and Mana recovery

By destroying an object or killing an enemy you can obtain items, gold or Recovery orbs. Recovery orbs recover HP or MP.
If a recovery orb is used while HP/MP are low some will be recovered immediately.
If your HP/MP is full a potion will be added to your inventory instead.
Try using an orb to fill your health now.

Trigger: Interact with orb

Figure 227. Using a recovery orb in *Legendary Tales*

As shown in Figure 227, the orbs are placed at a convenient height for the hand to touch them. The player needs to press a trigger to interact with the orb. This simple interaction has high levels of simulation and efficiency. In addition, the game design adds the efficiency value: these orbs can be obtained by destroying objects or enemies to sustain the gameplay; if the health or mana is full, the orb will become a potion added to the inventory. This design makes me think of the potion metaphor, and the orb skips the drinking interaction to recover the health more efficiently.

In *SWORDS of GARGANTUA*, no button pressing is required to use the orb, and players simply touch it to recover health over time. Given the nature of sword-fighting combats, I still find the one-time use is more efficient than the healing over time because standing still in a position to keep touching the orb can be vulnerable to incoming enemies. There is little or no difference in their simulation value.

Figure 228. Touching an orb to recover health points in *SWORDS of GARGANTUA*

7.6.3. Heal and Rescue

SWORDS of GARGANTUA implements a similar control for rescuing teammates by touching an icon. However, I feel it does not have the same simulation value as the recovery orb because the verb "rescue" has more embodied meaning than touching an icon. Also, the icon does not look magical or hyper-realistic enough to get away with the abstraction. In my experience with the 2D icon, I found it did not work smoothly as the 3D orb. Sometimes the progress bar moved then stopped, even I penetrated my hand through it. I ended up moving my body more closely onto it to get it to work.

Figure 229. Touching a 2D icon to rescue a teammate in *SWORDS of GARGANTUA*

Although this section is not under the FPS combat interaction chapter, I include some FPS examples to show how the game context affects the choice of tools for recovery interactions.

Population: ONE uses a more embodied tool for rescuing interaction. As shown in Figure 230, the game introduces a pair of defibs that every player has in their inventory. To revive a teammate injured and knocked down by an enemy, players rub them together for a short amount of time to charge them. The faster they rub, the sooner the charging process will end. Once the defibs are fully charged, they show visual feedback of electricity. Players touch the teammate with the defibs to revive them.

(a) (b) (c)

Figure 230. In *Population: ONE*, players can charge the defibs by rubbing them together to revive a teammate. (a) the charging process; (b) the fully-charged state; (c) players are vulnerable when caught during the process.

Although the defibs are seemingly unrealistic, this interaction achieves realism and a high simulation value with the meaningful and narrativized interface. The charging takes time depending on the rubbing speed; however, absolute time is not the only factor to judge its efficiency. The interaction deliberately takes time in the gameplay. Furthermore, it gives out sound cues, so it becomes a high-stake, high-reward behavior, allowing the opponent to capitalize it by attacking the player who is reviving a teammate.

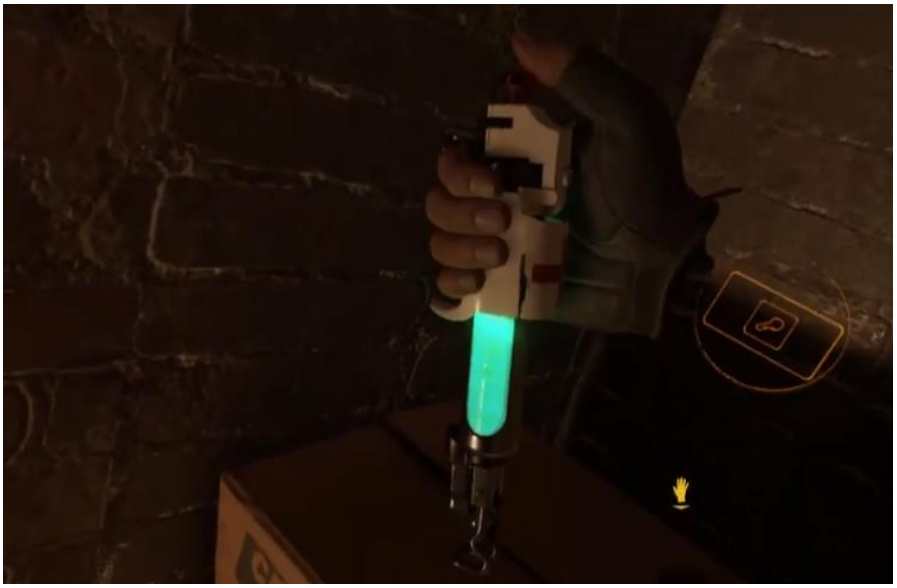


Figure 231. In *Half-Life: Alyx*, players can use a syringe to heal themselves by poking it into any part of their body. Screenshot from https://www.youtube.com/watch?v=CwV1CozegdQ

Half-Life: Alyx provides the player with two ways of self-healing in different scenarios. The first is a syringe that can be stored in the wrist pocket and used anytime in the gameplay. To apply it, players press the "A" button and poke it into any part of their body, as shown in Figure 231. This design simulates the realism of injecting mystery medical liquid into the body to recover health. The physics of the liquid adds to the simulation value. It has natural fluid properties and reduces brightness after being used.

Another way of self-healing in *Alyx* is to use a healing station found in the environment. As shown in Figure 232, the station costs the player a bottle containing a disgusting worm (not shown in the image) and squashes it into liquid. Then the player puts their left hand in a surgery interface and watches the robotic manipulators do the "healing" that may be painful to watch. In the later discussion, I will relate such experience to mirror-touch synesthesia.

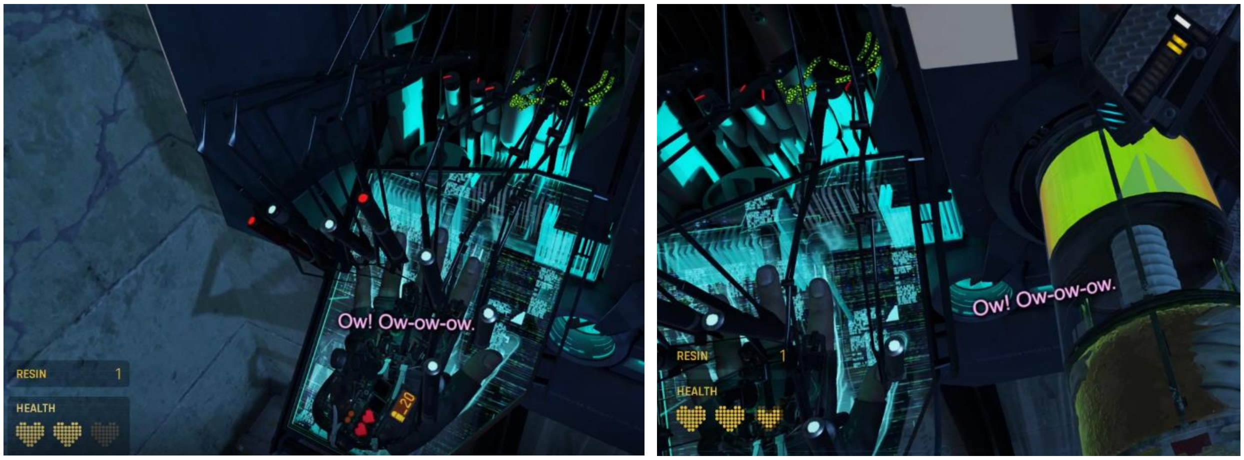


Figure 232. Using the health station in *Half-Life: Alyx*

7.6.4. Summary

In summary, the examples have overall been distributed in the top-right realm of the framework, as shown in Figure 233. The two outlines use 2D UI interactions that are less meaningful and natural than embodied interactions.

Among the rest examples that achieve moderate to high levels of simulation and efficiency, the simulation value is mainly revealed by the level of details in behavioral mimicking. The four examples with the highest simulation value feature a meaningful and functionally narrativized interaction and interface that reproduce the real-world behavior. The efficiency value is mainly revealed by the level of easiness to perform the interaction.

However, these two criteria can further cause tradeoffs between the two design values. The example of eating a banana by peeling four slides shows how simulating the detailed operations of the complex object structure can result in a higher error rate and affect its use in the time urgency context. The recovery orbs are metaphors of quickly consuming health or mana potions. Although their forms are unrealistic, they still achieve realism or "make sense' in the game context. The rest eating and drinking examples also vary their values based on the two criteria, but their variances of the value tensions are relatively small.

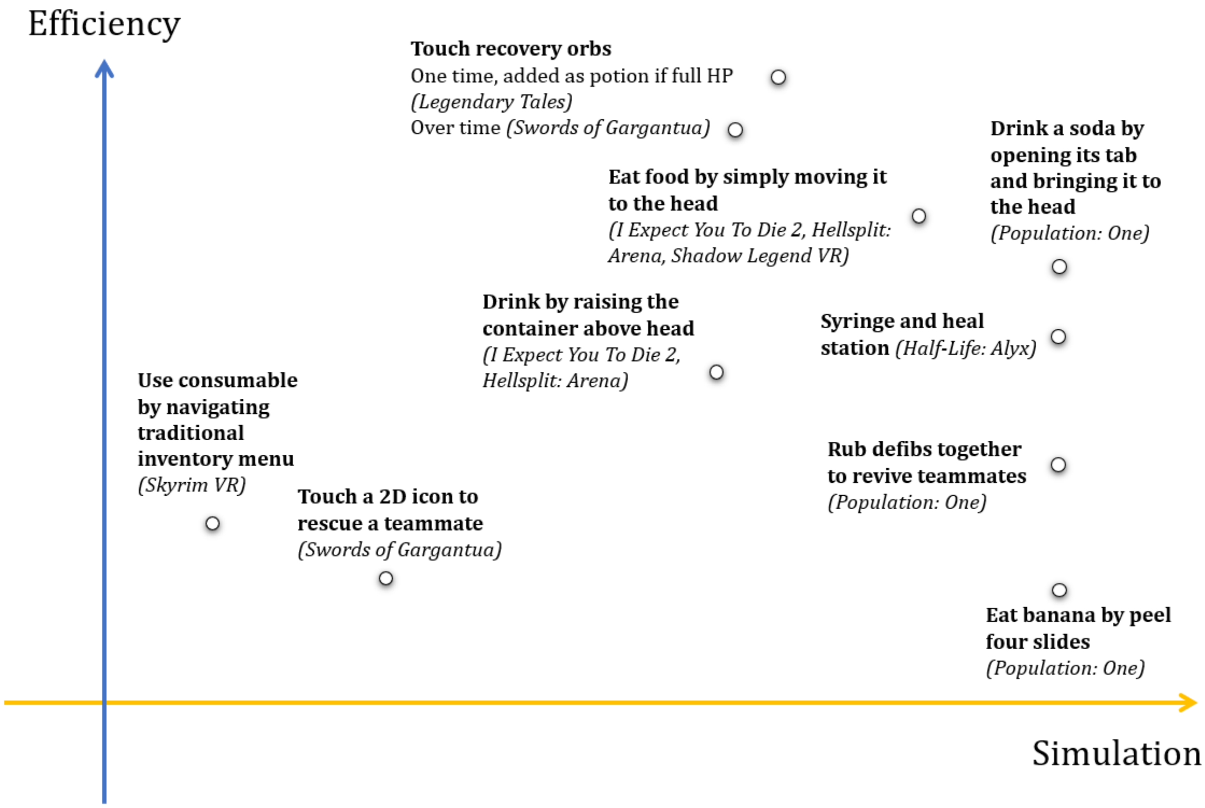


Figure 233. Summary and comparison of consume and recover interaction examples using the simulation and efficiency framework

8. Discussion and Conclusions

This dissertation contributes a design theory that frames the complex design space of VR game interactions with the two design values – simulation and efficiency – as its two primary dimensions. My case studies and close readings have mapped hundreds of examples onto the simulation and efficiency framework. The examples and analysis prove that these two fundamental design values are useful analytical lenses and design schemas for VR game interactions, which ubiquitously reveal value tensions between simulation and efficiency at different levels. They further show how the forms within the interaction, the game context, and the platform constraints reveal, amplify, and reconcile the value tensions. With the examples and findings, I theorize the design of VR game interaction as reconciling the value tensions between simulation and efficiency.

I approached VR game interactions from a formal perspective, breaking them into systems of locomotion, object, and combat interactions, and further decomposing an interaction into the action semantics, forms and feedback, controls and techniques, game contexts, and platform affordance and constraints. Still, when identifying and comparing their design values, I preserve a level of subjectivity. I provide explicit thought processes and specific reasons in the analysis regarding what criteria I rely on when making any judgement calls.

There is no clear answer for which design case works the best or any golden rules guaranteeing the best design in this space. Designers can always refer to the respective section for a specific interaction to find cases that best suit their needs. Nevertheless, with all the examples and the analytical work I have done, I see opportunities to formulate some theoretical and practical contributions as take-aways for scholars and designers interested in this field.

8.1. Theoretical Contributions

In this section, I will revisit the analytical lenses of simulation and efficiency to synthesize the new understandings that we have achieved from the analysis. I will revisit the theoretical aspects of my research questions and discuss how the new understandings of efficiency and simulation as design values for VR game interactions can extend the existing design theories and literature with which I start this research.

8.1.1. New Understandings of Simulation and Efficiency

When I started this research, I had inspirations of using design values and value tensions in HCI and game studies as the lenses to understand the design space of VR game interactions. I had some vague ideas of what simulation and efficiency mean in this context but was unsure whether they would be as productive as expected, or they were just placeholder concepts that I would replace with better terms to further refine them. However, the more I have analysed the examples and reflected on the existing literature, the more confident I feel in using them as meaningful and “authentic” design values to examine the interactions and reveal gaps in our current understandings.

8.1.1.1. Simulation – the experiential and realism over the formal and realistic

The most important new understanding of simulation that I want to highlight here is how significant the experiential perspective of simulation is in VR game interactions over the overly-discussed formal perspective. *Feeling* an interaction is real in VR games is more important than noticing it is technically realistic in every detailed form and function.

Within game studies, I continue to find many discussions around games and simulation, that in some ways, focus on the formal perspectives of simulation, such as technology, systems, computing power, modeling, accuracy, flight simulators, military and war (Crogan & Aarseth, 2008; Wark, 2009; Crogan, 2011). A historical and technocultural account for this emphasis could be that modern computers brought the renaissance of video games able to simulate high-quality 3D worlds and large-scale events. These discussions, together with the earlier ones I listed in section 2.1. focus on the ontology and the potential of video games as computer simulations.

However, the formal simulation perspective can create a realistic experience, but not always, and can even ruin it if only a small part does not feel right. I focused on the formal elements when examining the simulation value, but in my case studies and close readings that have come across many different forms, I find that some have much higher weight than others in making me believe that the interaction is of a higher simulation value.

For example, in the climbing interactions (see Figure 234, or Figure 87), the highest simulation cases do not have realistic simulations of the arm or the wrist; they do not even have a body, only the hands, but everything looks and feels normal. The cases with lower simulation values have additional formal elements that arguably strive for more realistic physics. However, they turned out to be uncanny in some ways in my experiences. For instance, I felt the game's immersion was broken when moving my body into the space between two bars when climbing the ladder, moving my hands into the wall, or seeing visual distortions and mismatches. These “butterfly effects” affected how I thought they were good simulations. In contrast, I did not feel that not having realistic body simulations was an essential factor in simulation, as long as the central action – climbing - had achieved *realism*.

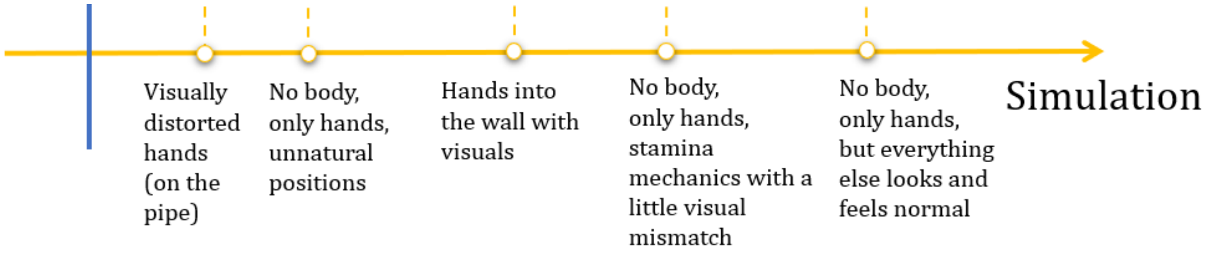


Figure 234. Formal elements with more realistic physics features in climbing end up being uncanny and lowering the experience of simulation

A similar argument can be made with reloading interactions (see Figure 235 or Figure 158). I did not feel that the standard “snapping” of the supplementary clip into the gun was an inaccurate simulation at all. The more significant course of interaction that overweighted this minor unrealistic abstraction is my hand grabbing the clip and moving it to the gun. The absolute precision of where I released it was not a critical factor. In contrast, the realistic right handle misdirected me to experience low simulation. It forced me to pull it with my left hand instead of how it should probably be pulled with the right hand in the real world; however, in the VR gaming context, the right hand always needed to hold the gun. Besides, the non-diegetic visuals appear as eye-catching forms when everything else is realistic, and they negatively affected my sense of simulation value as well.

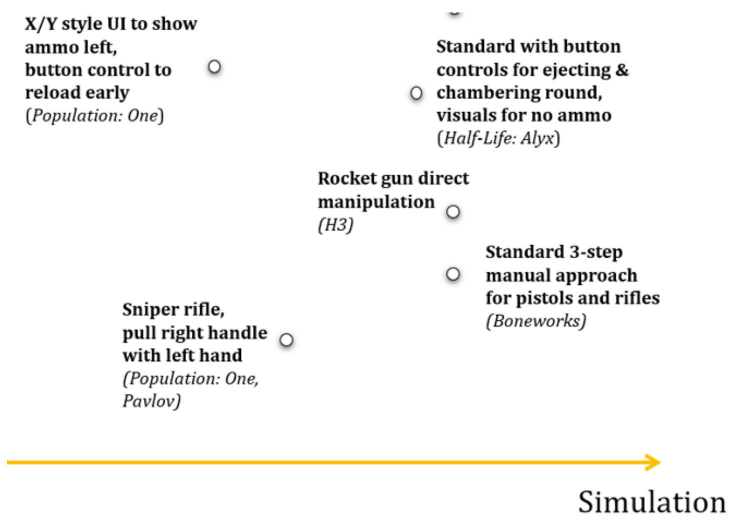


Figure 235. The simulation value of reloading interactions is minorly affected by the snapping mechanic in the standard approach, but the realistic positioning of the right handle and the non-diegetic visuals more significantly affect my experience of simulation.

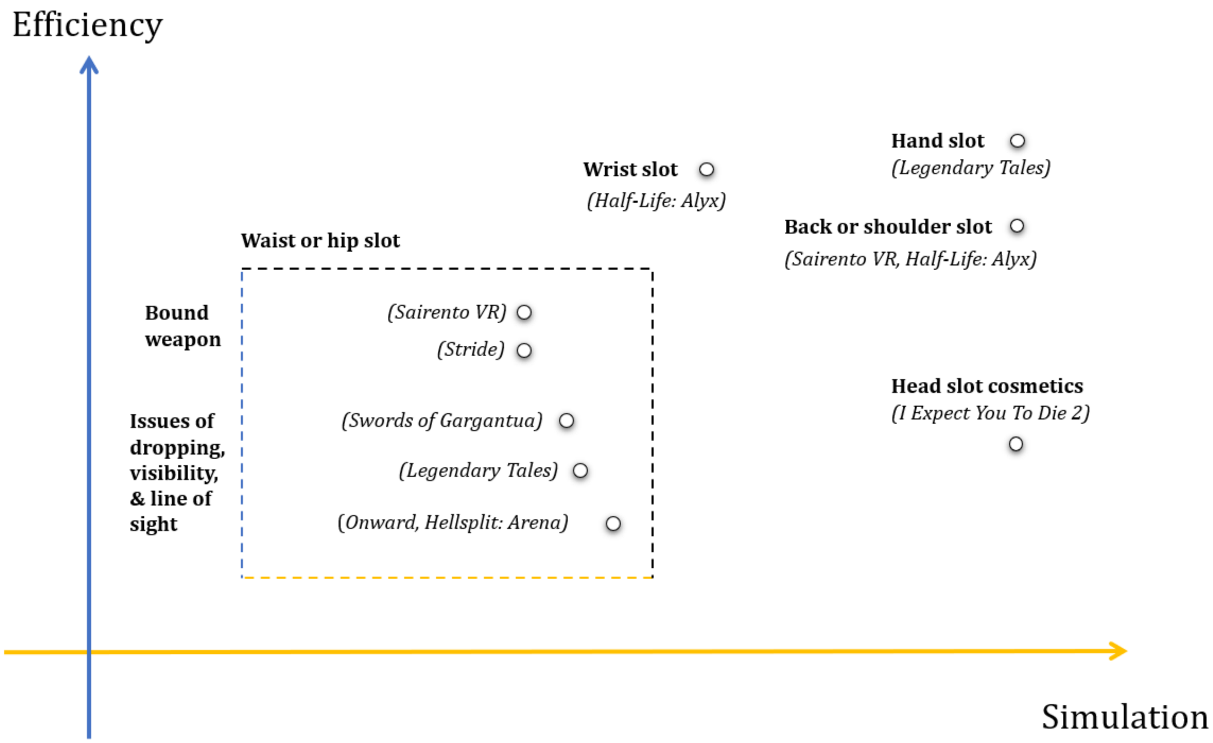


Figure 236. All the body slots are realistic simulations in terms of their positions; however, not all of them deliver the same experience of “they actually exist there or are easy to access.”

The body slots also support the experiential perspective. As shown in Figure 236, all the body slots are realistic in terms of their positions, at least semantically. However, not all of them deliver the same experience of “they actually exist there or are easy to access.” The poor experience of simulation partially comes from efficiency issues, which are due to attempts of being more realistic, such as estimating slot positions far from the tracked devices, and realistically allowing the weapon to drop, and blocking the line of sight of the waist slots with the chest armor.

Consider experiencing an interaction in VR games as participating in a *magic show*. The goal is to use any tricks to fool the audience that the magical phenomena they see are real. The tricks, or the internal truth that only the magician knows, are like the formal perspectives of simulation. They can hide within the realistic experience, which is all that matters to the audience: the external truth, the representation, or the illusion. Therefore, the new understanding of simulation for VR game interactions is about making players believe they are realistic simulations.

Besides the emphasis on the experience, there are other new understandings that can be derived from this perspective and the examples to add to the theoretical perspectives of simulation as a design value.

Diegesis is a factor that may also affect the experienced simulation, especially when seeing a non-diegetic virtual form in a realistic interaction and environment. Previously, I did not think diegesis was related to simulation. However, in my case studies and close readings, I have come across countless cases where I rank an example to have lower simulation value because it has non-diegetic visuals, such as UI icons, texts, numbers, and menus. Of course, these non-diegetic elements may be efficiency-oriented. Still, they can immediately break my immersion and belief, making me question whether they are realistic or not. This potential value tension can be amplified if the non-diegetic forms do not fit with the overall theme created by other forms. It can be reconciled if they do fit or if the efficiency value significantly boosts the gameplay.

As we have noted, the experienced simulation may also incorporate efficiency to various extents. The high-simulation examples tend to reproduce the most important, meaningful, defining phase of a behavior, leaving the less significant steps as abstractions to avoid losing the simulation value from potential problems of over-stimulation or being too realistic. In this sense, they are efficient simulations.

Finally, the interaction does not necessarily simulate a physical counterpart. As we have seen from many hyper-realistic examples, such as teleportation, combat abilities, and futuristic technologies, what the interaction simulates can be an imaginary concept or a fictional object's action or function. Also, some VR game interactions can be understood as second-order simulations, which do not directly simulate their physical counterparts, but the reimaginations or adaptions of the physical.

With all these new understandings of simulation, I reconceptualize simulation as a design value with the following definition to incorporate the experiential perspective in addition to the formal:

The design value of simulation is the extent to which the (VR game) interaction achieves realism by making the player experiencing the interaction believe it is a realistic representation of a physical or imaginary counterpart. It can be evaluated through the player's experience and achieved with realistic forms, behavioral mimicking and metaphors, and interaction fidelity.

8.1.1.2. Efficiency – a hierarchical model

Efficiency was defined as an adaption of *ludic efficiency* (Tanenbaum & Bizzocchi, 2009) to incorporate the interaction rather than the original focus on the interface:

Efficiency is the extent to which the interaction design eases or hinders the player's attempt to perform the game action.

I also incorporated usability (International Organization for Standardization, 1998), playability (Järvinen et al., 2002) and game flow (Sweetser & Wyeth, 2005) aspects, such as execution time, input and output mechanism, cognitive workload, error rate, game goals, and feedback. Overall, these efficiency aspects remain important and useful in my close reading to evaluate the interaction examples. Still, I want to add a few new ideas from the VR platform and game contexts that help understand efficiency as a design value.

I present a hierarchical model to list the priorities of the related efficiency aspects. First and foremost, I contend some baselines of *immersivity* should be prioritized to create an inclusive and engaging experience. If an interaction makes players feel uncomfortable over a sufficient period of time during the gameplay, it will prevent them from engaging with the content and immediately break their immersion. Examples include motion sickness, limiting the sight or field of view, distorted bodies, body parts into rigid surfaces, intense and sharp visuals hitting the eyes, such as projectile traces and tips of weapons, and the mirror-touch synesthesia in violent, blood and gore content (discussed in section 7.2). These sensory stimuli are not the core mechanics or meaningful game content but cause extra cognitive burden and discomfort for players, making them want to close their eyes or escape from their headsets. These are the most severe problems of efficiency and should be avoided. However, many of these sources of inefficient gameplay are not purely issues arising from the hardware limitations but the software design.

Above the baselines of ensuring immersivity and sensory stimuli, the next priority is avoiding *easy-to-fail* interactions. These interactions pull the players away in the opposite direction from achieving the intended game goals, leaving them in disadvantages. Examples include unbounded weapons and guns accidentally dropping in combat, failed reloading, failed body slots or grabbing a wrong item, and failure in peeling four slides to eat a banana. These examples typically result from unhandy physical interactions and the strict trigger conditions that ask for a high level of precision in those physical interactions. The easy-to-fail interactions may not be the most important game mechanics, but they are usually an important step within it, and failing them causes the player to be at a disadvantage position or waste a significant amount of time. As a result, they end up with an experience that punishes the player for their human factors, making them feel stupid in repeating trivial activities and unable to engage with the core gameplay. Designing easier game mechanics and more unhandy controls can potentially resolve the problems of efficiency at this level.

Above the "easy-to-fail" level, the next level is *physical inefficiency*. Again, some interactions may fall into the easy-to-fail cases. But for others that have a low failure rate they can cause fatigue and inefficiency over a long time. Examples include physical hand swinging for running, physical crunching, physical dodging, etc. These interactions may have a high or partial simulation value of behavioral mimicking and interaction fidelity, revealing the tension with efficiency. More broadly speaking, they are the ones that lead to the inequality among VR players, with those who are more physically capable and have a large open space for room-scale physical activities being more qualified and competent to play than those who prefer the seated mode.

Although this hierarchy can continue in more detail, I conclude the rest examples to be at the *contextual efficiency* level, which is the top level. Interaction examples at this level incorporate the related aspects of efficiency in various ways. Contextual means the value is evaluated based on how well the interaction is efficient in the game context. It avoids using absolute time as the primary criterion, which seems to be the dominant view in our common sense and usability studies of game interactions (Krompiec & Park, 2019). Instead, the contextual efficiency examples include using simple controller inputs, providing feedback and visual aids to facilitate information processing, leveraging gestures and affordances of the platform, optimizing mechanics and interaction techniques, and reusing control schemes. An interaction that takes longer is not necessarily inefficient.

I summarize the above discussion in Table 2 to present the hierarchical model of efficiency.

Table 2. The hierarchical model of efficiency as a design value for VR game interactions

Efficiency Level	**Description**	**Examples**	**Related Concepts in Literature**
Contextual Efficiency	Interaction is efficient in the game context and facilitates players to achieve game goals.	Simple controller inputs, feedback and visual aids to facilitate information processing, leveraging gestures and affordances of the platform, optimized mechanics and interaction techniques, reusing control schemes, etc.	Gameflow aspects: cognitive workload, game goals, feedback (Sweetser & Wyeth, 2005), Usability aspects: time and learning curve (International Organization for Standardization, 1998), Ludic efficiency (Tanenbaum & Bizzocchi, 2009)

Physical Inefficiency	Interaction leads to fatigue and requires the player to have the physical capability and a large open play area. They may have a high simulation value but are not rewarding.	Physical hand swinging for running, physical crouching, physical dodging, realistic swimming, etc.	Interaction fidelity with high symmetry in body motion body segment, and control function (McMahan, 2011), Behavioral mimicking (Bizzocchi et al., 2011), Narrative and kinematic perspectives of embodied game interfaces (Tanenbaum & Bizzocchi, 2009)
Easy-to-fail Interactions	Interaction is easy to fail and prone to error. It deviates the player from achieving the intended game goals, leaving them in disadvantages, repeating trivial activities, and being unable to engage with the core gameplay.	Unbounded weapons and guns accidentally dropping in combat, failed reloading, failed body slots or grabbing a wrong item, failure in peeling four slides to eat a banana	Gameflow aspects: control, game goals (Sweetser & Wyeth, 2005), Ludic efficiency (Tanenbaum & Bizzocchi, 2009)
Poor Immersivity	Interaction immediately breaks the immersion or delivers painful sensory stimuli and content that	Motion sickness, limiting the sight or field of view, distorted bodies, body parts into rigid surfaces, intense and sharp visuals hitting the eyes, such as	Motion sickness (Reason, 1978, pp. 819–829), Mirror-touch synesthesia (Banissy & Ward, 2007) Functional, structural, and

	make players want to avoid or escape from VR.	projectile traces and tips of weapons, mirror-touch synesthesia in violent, blood and gore content.	audiovisual playability (Järvinen et al., 2002)

With the new understanding of efficiency, I update its definition to incorporate the priority of achieving the goal in the game context.

The design value of efficiency is the extent to which the interaction design eases or hinders the player's attempt to achieve the desired goal in the game context. The goal can include performing a game action, but performing a game action does not mean successfully achieving the desired goal with the action.

8.1.2. Revisiting Research Questions and Extending Design Theories

I opened this dissertation with two research questions that have structured my inquiry in this work. Now it is time to revisit the theoretical perspectives in my research questions and discuss how my work extends existing design theories.

> The first question comes from applying examples of VR game interactions to explain and understand existing theoretical frameworks on the concepts of simulation and efficiency. It seeks to extend the frameworks in Narrative and Embodied Interface and Interaction Fidelity by considering the duality of the two design values:
>
> **RQ1:** How do VR game interactions reveal value tensions between simulation and efficiency?
>
> The second question comes from applying the new framework of simulation and efficiency as lenses to explain and understand the design space of VR game interactions:
>
> **RQ2:** What design strategies for VR game interactions can we learn with the new understanding of simulation and efficiency? Specifically, in what ways and to what extent do VR games reconcile or amplify the tensions between simulation and efficiency?

For these two questions regarding how VR game interactions reveal the value tensions between simulation and efficiency and the ways and the extent of reconciling or amplifying them, my case studies and close readings from chapters 4 to 7 have addressed them in detail for each semantic game action. Therefore, I will discuss these questions with everything I have learned so far at a higher level.

First of all, as we discussed in the previous section, the design value of simulation can involve the concept of efficiency, and the design value of efficiency situated in the context can also involve simulation. Simulation and efficiency are not two independent variables in their nature, although I contend they are still useful lenses when viewed separately. Simulation is how the interaction achieves realism, not necessarily to maximize every information and technology to be realistic.

Now consider two interactions identical in every aspect and have both achieved realism, except one has the information and technology to be more realistic in some way. As we have seen in some examples in the current state of VR games, a little difference in being more realistic may cause an inefficient or uncanny experience, which can break the feeling of simulation. The phenomenon of a small increase in the realistic direction beyond achieving realism makes the experience uncertain is akin to the concepts of singularity[29] and the uncanny valley effect.

This suggests a different focus on the uncanny valley model than how it is applied to interaction fidelity (McMahan et al., 2016). The current framework of interaction fidelity is essentially an abstraction of the uncanny valley model by focusing on the valley part of the curve and its overall “high-low-high” tendency, as shown in Figure 237. The authors claim that the effects of interaction fidelity on user performance and experience are like a U-shaped curve. As a comparison, my theory emphasizes the V-shaped uncanny valley as a potential singularity in the simulation experience, a point after the interaction has achieved realism and continues to strive for being more realistic, as shown in the right image in Figure 237.

---

[29] The term singularity has a meaning of a point in time where a growing technology, such as robots and aritificial intelligence, becomes uncontrollable and irrevserbile, leading to unforeable changes to human civilization (Kurzweil, 2005). In other fields of system science, it can mean a point in the simulation where the system’s dynamics becomes unforeable, uncontrollable, or chaotic.

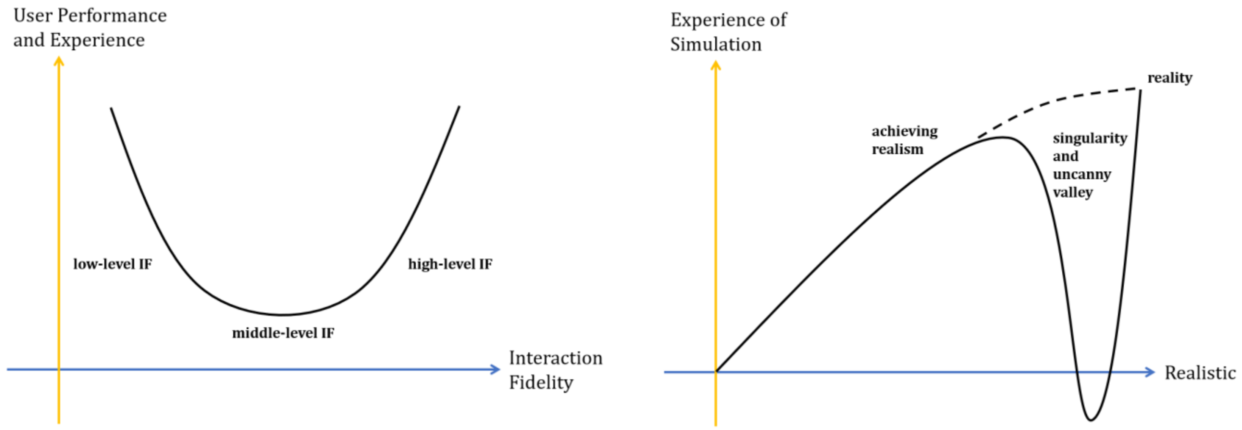


Figure 237. Two different perspectives on the uncanny valley model. The interaction fidelity framework focuses on the "valley" (left). My simulation model focuses on the "uncanny" (right).

I contend that for the interaction to achieve realism without being too realistic is an efficient simulation. From this point, the interaction inherently reveals the value tension between simulation and efficiency. Depending on the level of simulation when approaching realism and the level of efficiency, the interaction can be an optimal design that reconciles the value tension and achieves both high values, or it can still be suboptimal in either or both directions.

However, the focus is on the possible singularity and the uncanny valley after the interaction has achieved realism. The value tension can be amplified by implementing a slight increase for the system to be more realistic, but with an unexpected chain effect from other behaviors that drastically drops the simulation experience. It implies that the higher we want the interaction to be realistic, the more effort we will have to make to ensure its performance and stability in delivering the simulation experience.

From the efficiency perspective, we can identify value tensions at each level in the hierarchy of efficiency (Table 2). The levels of poor immersivity and easy-to-fail interactions reflect how being overly realistic can fall into the uncanny valley and lead to disasters and failure in the experience. The level of physical inefficiency reveals the tension between simulation and efficiency from the unrewarding physical activities. If the interactions are mundane locomotion interactions, in many cases, efficiency will need to be prioritized. At the contextual efficiency level, the value tensions can yield mixed results, and that is where we can find many optimal examples and why part of interaction design remains an art of design.

The ubiquity of the two design values and their tradeoffs among VR game interactions also informs the similar concepts in the frameworks of Interaction Fidelity (McMahan, 2011) and the Narrative and Embodied Interface (Tanenbaum & Bizzocchi, 2009). Because I have already repeatedly discussed some key ideas of extending them throughout the dissertation, I will formulate the points for concise to close my theoretical contributions.

Extensions to the theoretical perspectives in the Interaction Fidelity framework and the Narrative and Embodied Interface framework:

1. Extending the dimensions and the model

The Interaction Fidelity (IF) framework is a single-dimension approach to describing the design space of VR interactions. The Narrative and Embodied Interface framework provides three useful prospectives: ludic, kinesthetic, and narrative, but is limited in further investigations into their interconnections. My study further explores the relationships between the design values of simulation (mapped to high IF and the narrative perspective) and efficiency (mapped to the low IF and the ludic perspective) to better understand the design space of VR games. Specifically, my study rebuts the "middle level IF is the worst" statement and provides a new perspective on the uncanny valley model of IF.

2. Extending the referents and criteria

The two frameworks both emphasize "realistic" and "physical" behaviors as their referents in their formal definitions of simulation concepts. My investigation into the design space of VR games shows that being too realistic can amplify the value tension between the experienced simulation and the contextual efficiency and that an imaginary counterpart can also be the referent of simulation. In addition, my study extends the criteria for simulation and efficiency by taking well-established formal concepts from the two frameworks and considering additional factors from the VR platform and game interaction contexts.

3. Extending the scope

The IF framework has been chiefly applied to VR game prototypes and comparisons between a limited number of cases, such as (McMahan et al., 2012; Nabioyuni & Bowman, 2015; Rogers et al., 2019). The narrative and embodied interface framework was established from a music simulation game (*Rock Band*) with specialized tangible interfaces (Tanenbaum & Bizzocchi, 2009) and further applied to games on desktop, embodied, and mixed-reality platforms (Bizzocchi et al., 2011). This study focuses on a state-of-the-art commercial VR platform (Oculus Rift S) and two popular game genres (FPS and RPG) but covers hundreds of interaction examples in twenty categories of semantic actions. My data-driven analysis shows the formal criteria of interaction fidelity, ludic efficiency, functional narrativization, behavioral mimicking and behavioral metaphor still apply and yield productive results in identifying and comparing the design values of simulation and efficiency. Furthermore, the focus and scope extend the discussions in (McMahan et al., 2016; Rogers et al., 2019) and further articulate and diversify the effects of IF for different kinds of actions and game contexts. The systematic analysis of the dataset updates our understandings of these formal criteria by revealing their value tensions and significance in designing each kind of game action under a specific context.

Based on these extensions, I suggest that my new theory of simulation and efficiency is a more comprehensive framework that incorporates the two existing frameworks to understand and explain the design space of VR game interactions. Next, I will summarize the practical contributions of my study.

8.2. Practical Contributions

First and foremost, let me promote my VR Game Interaction dataset and associated case studies as a substantial reference for interaction designers and developers interested in building games, simulations, or metaverses in VR. I have taken a brutal-force approach to "hack" the most popular commercial VR FPS and RPG games and break their interaction systems into modules of more than seven locomotion interactions, three object interactions, five FPS combat interactions, and five RPG combat interactions. These modules of game actions are a collection of "verbs" or game dynamics for designers to mix and match.

Under each module are further variations of the action in its interaction techniques or components. You will see five to ten well-explained screenshots for each action to show the most representative design approaches. These screenshots are carefully taken and selected from a pool of high-fidelity gameplay recordings to capture the interaction design's essential forms, techniques, and context.

At the end of each interaction module, I summarise and compare the interaction examples using the simulation and efficiency framework. Then, I explain which ones I think are the optimal design cases that achieve high simulation and efficiency values, which ones I experienced value tensions within them, and how they amplify or reconcile the tensions. Based on my evaluations, designers can decide which design cases suit their needs the best.

This systematic, comprehensive, and data-driven review is inspired by well-received books, academic papers, and datasets on game interaction design. For example, Katie Salen and Eric Zimmerman's book *Rules of Play: Game Design Fundamentals* includes a collection of design schemas showing how games can be modeled as different systems or approached from various experiential and cultural values (Salen & Zimmerman, 2003). Mary Flanagan and Helen Nissenbaum's book *Values at Play in Digital Games* is another inspiration for this work to follow their heuristics of discovery, implementation, and verification of design values (Flanagan & Nissenbaum, 2014). In the field of VR interactions, Jason Jerald's *The VR Book: Human-Centered Design for Virtual Reality* has laid some fundamental concepts and patterns for basic viewing, selection, and manipulation interactions (Jerald, 2015). Smaller-scale studies are evaluating VR interaction techniques, 3D user interfaces, and their design values such as naturalness, high fidelity, immersion, presence, and efficiency (McMahan, 2011; Bowman et al., 2012; Nabiyouni et al., 2015; Nabioyuni & Bowman, 2015; Frommel et al., 2017; Seibert & Shafer, 2018; Krompiec & Park, 2019; Rogers et al., 2019; Sweetser et al., 2019; Weech et al., 2019; Ng et al., 2020). With the prevalence of commercial VR, recent studies have started to uncover more practical design knowledge in specific categories, such as locomotion and social interactions, by analyzing cases and datasets from commercial VR games (Tanenbaum et al., 2020; Al Zayer et al., 2020; Di Luca et al., 2021).

Therefore, to practically situate my work within the above inspirations and movements, I extend the sparks of fire and develop the value lenses in the field of VR interactions to illuminate a broader and more complex design space presented by the state-of-the-art of commercial VR games. I find this design space has richer game contexts than the early prototypes, a more diverse range of interactions beyond locomotion systems, a standard gaming platform to normalize the variable of physical affordances and constraints and a more coupled pair of fundamental design values. The design knowledge that I uncover fills the gap of not having a comprehensive understanding of these unique aspects of the new design space, which can only be reached through a closed examination of the interaction forms and experience within a large number of thoughtfully chosen examples.

The case studies have already detailed the value tensions and approaches to balance them for each action. In the rest of this section, I will present some meta-analyses based on the results from all the case studies and formulate some guidelines for using the simulation and efficiency framework.

8.2.1. Emerging Best Practices and Representative Examples

The number of data and examples reviewed in this dissertation allows many interesting ways of meta-analysis and presenting the results, including mapping the optimal cases, the average point, the variance of each interaction category, and further comparisons among different categories. However, to keep the scope, I present the emerging best practices, or in theory, the optimal cases from each interaction category. In addition, the most representative cases that fall onto the other corners of the simulation and efficiency framework will also be presented together for comparisons.

8.2.1.1. Locomotion Interactions

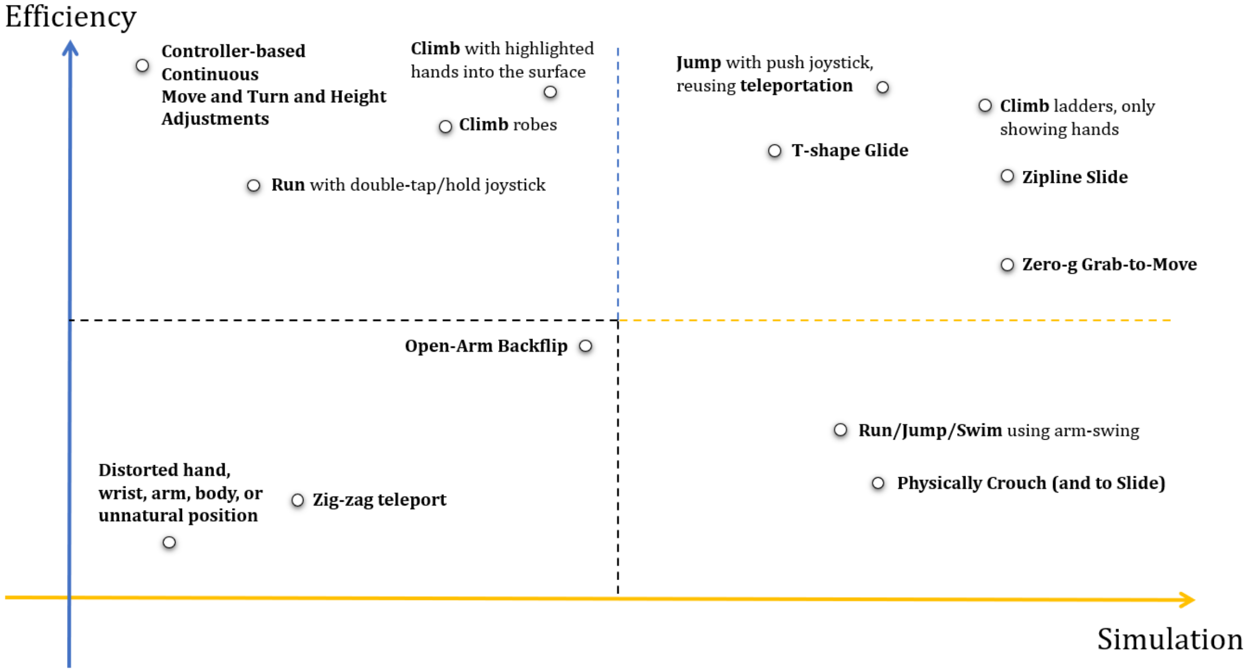


Figure 238. Representative locomotion interactions: the optimal cases, the high-efficiency-low-simulation cases, the high-simulation-low-efficiency cases, and the low-efficiency-low-simulation cases.

For better clarity and a visual presentation, I formulate the discussion into the following tables.

Table 3. Optimal (high-simulation, high-efficiency) locomotion interaction design cases

Optimal Locomotion Interaction Design Cases	References in Case Studies	Description
Climb ladders, only showing hands	*Half-Life: Alyx* (Figure 85)	The interaction feels steady and achieves the goal—no distortion or unnaturalness, auto-lifting at the end. Not showing arms or the body does not lose much of the realism.
Jump with push joystick, reusing **teleportation**	*Sairento VR* (Figure 69), *Half-Life: Alyx* (Figure 68)	The examples combine jumping with teleportation to shift the hard-to-simulate physical jumping to a hyper-realistic action in VR. In addition, the multiplexing of joystick control is efficient.
T-shape Glide	*Population: ONE* (Figure 76)	Open arms in a T shape to glide (fly) in mid-air.
Zero-g grab-to-move	*Echo VR* (Figure 88)	The interaction is a highly contextual case in zero-g, yet it is easy-to-perform and fits the environment.
Zipline Slide	*STRIDE* (Figure	The interaction relies on zipline affordances to

	75) *SwarmVR* (Figure 94)	achieve efficient locomotions and controls.

Table 4. High-efficiency, low-simulation locomotion interaction design cases

High-Efficiency, Low-Simulation Locomotion Interaction Design Cases	References in Case Studies	Description
Climb with highlighted hands into the surface	*Population: ONE* (Figure 83)	The interaction achieves contextual efficiency with non-diegetic highlighting visuals and a universal solution that turns any surface climbable.
Climb robes	*Legendary Tales* (Figure 82)	The interaction design does not simulate body physics on a robe but achieves contextual efficiency because of that.
Controller-based Continuous Move and Turn and Height Adjustments	Standard option in most VR games	The most used basic locomotions prioritize efficiency over simulation, with options to mitigate motion sickness. Height adjustments provide a useful crouch function for the seated mode.
Run with double-tap/hold joystick	*Boneworks* (Figure 50)	Double-tapping/holding the joystick as a button or in a direction or as a button supports easy controls for a multiplier to the basic movement.

Table 5. High-simulation, low-efficiency locomotion interaction design cases

High-Simulation, Low-Efficiency Locomotion Interaction Design Cases	References in Case Studies	Description
Physically Crouch (and to Slide)	*Boneworks* (Figure 61), *Sairento VR* (Figure 72)	The physical crouching is a direct mapping but is physically inefficient, especially for the seated mode.
Run/Jump/Swim using arm-swing	*Legendary Tales* (Figure 82)	Arm-swing is a good metaphor for physical running or jumping. It is a direct mapping of the hands in swimming. However, it leads to fatigue and does not reward the player much in the game context.

Table 6. Low-simulation, low-efficiency locomotion interaction design cases

Low-Simulation, Low-Efficiency Locomotion Interaction Design Cases	References in Case Studies	Description
Climb with distorted hands and unnatural body positions	*Boneworks* (Figure 80), *Legendary Tales* (Figure 81)	Distorted body parts and unnatural positions are poor simulation and fall into the "poor immersivity" level of efficiency. The design cases need to resolve or reconcile the conflicts in hand-tracking to move the full body in climbing, such as the rigidbody.
Open-Arm Backflip	*Sairento VR* (Figure 77)	This interaction is an interesting and arguable case. It can be efficient and fun in the game context but is prone to motion sickness.
Zig-zag Teleport	*Legendary Tales* (Figure 58)	This "comfort" option is an inefficient roundabout locomotion interaction technique, not useful in achieving any ludic goal, and a poor simulation of the teleportation concept.

Among the above results, we can see that the optimal locomotion interactions feature hand-based actions that achieve realism but are not the core movement mechanics that players have to perform repeatedly over a long time, which can otherwise cause fatigue like the high-simulation, low-efficiency ones do. In this sense, they achieve an optimal balance between the gameplay challenges and the player's physical skills. The most used movements prioritize efficiency over simulation. They ensure the easiness of progressing in the game space to see the new content rather than use the realistic embodied motion to keep the player in the immersion and the flow state.

8.2.1.2. *Object Interactions*

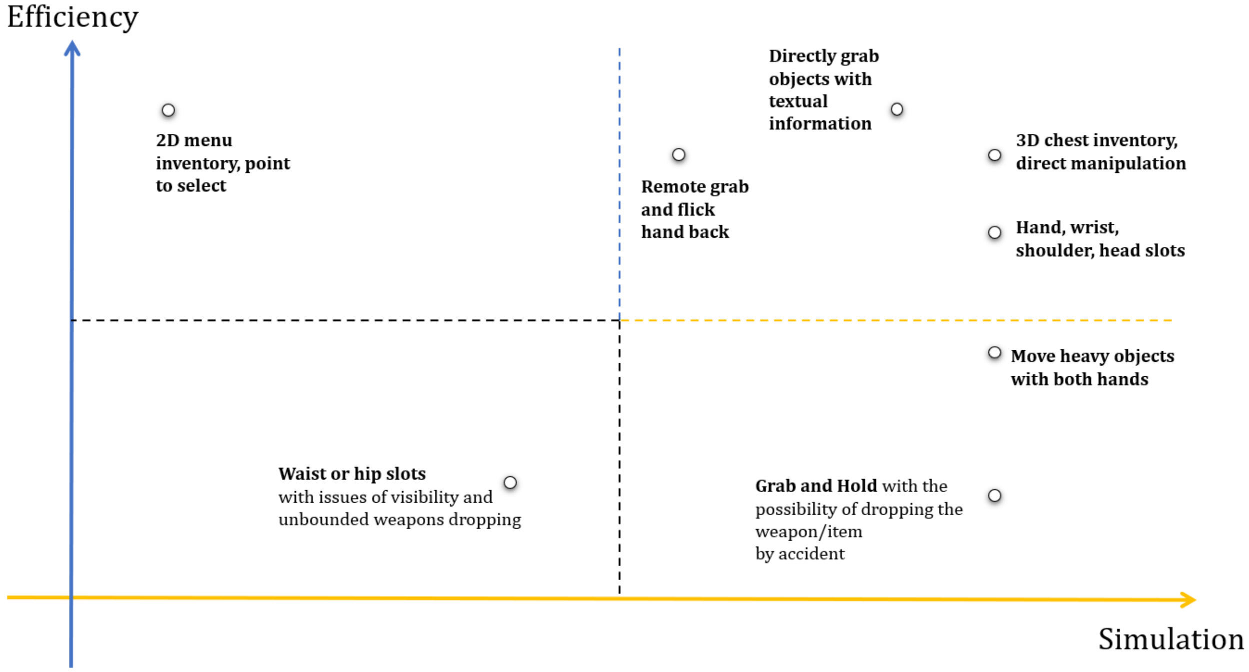


Figure 239. Representative object interactions: the optimal cases, the high-efficiency-low-simulation cases, the high-simulation-low-efficiency cases, and the low-efficiency-low-simulation cases.

Regarding the representative object interactions in Figure 239, the variance is much clearer than the locomotion interactions, and therefore I will not draw the tables to describe them in detail. The best practices feature direct manipulations with hands to grab objects in the environment and manage them in slots near the head and hands or embodied 3D inventory interfaces. Visibility and dropping are the two significant issues that can severely lower simulation and efficiency values, as shown in the suboptimal examples.

8.2.1.3. *FPS Combat Interactions*

Figure 240 shows the representative cases in the FPS Combat Interaction category, and the following tables provide extra descriptions using the concepts we have established. Various context-specific factors affect the variance in this space, including visuals, physics, mechanics, the weapon's affordances and constraints.

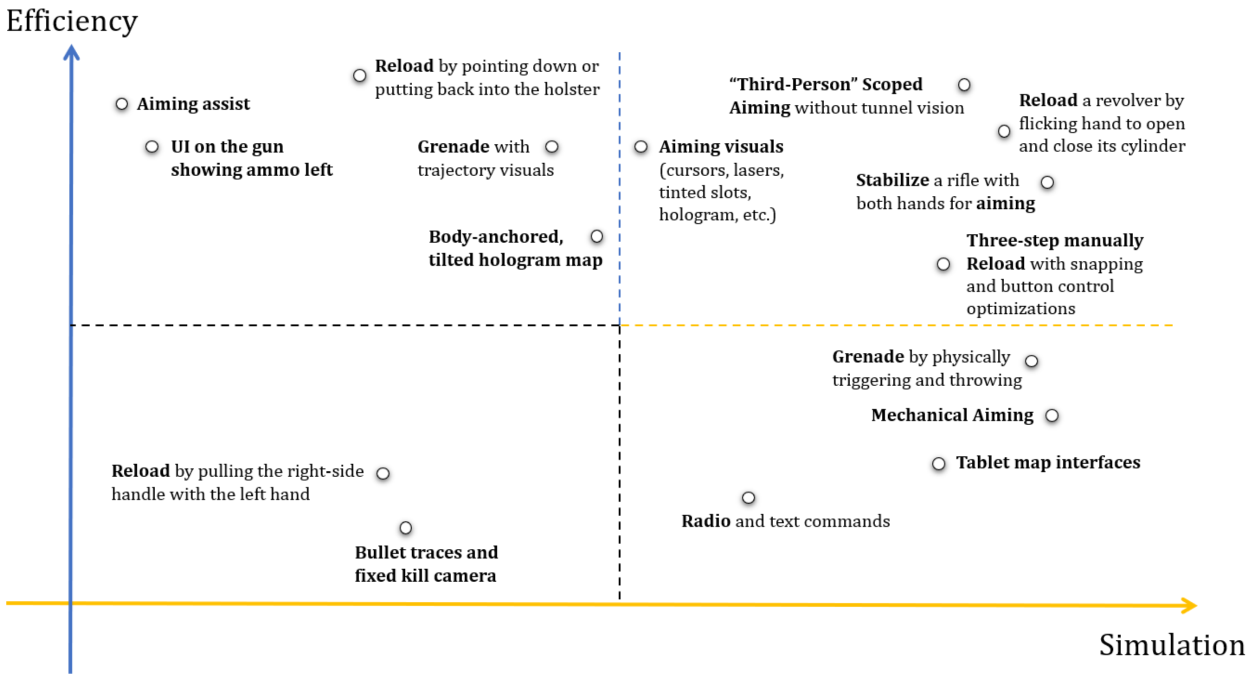


Figure 240. Representative FPS combat interactions: the optimal cases, the high-efficiency-low-simulation cases, the high-simulation-low-efficiency cases, and the low-efficiency-low-simulation cases.

Table 7. Optimal (high-simulation, high-efficiency) FPS combat interaction design cases

Optimal FPS Combat Interaction Design Cases	References in Case Studies	Description
Aiming visuals (cursors, lasers, tinted slots, hologram, etc.)	*Frostpoint VR: Proving Grounds, Population: ONE* (Figure 131), *Boneworks* (Figure 128), *Half-Life: Alyx* (Figure 129)	The visuals can be rationalized to be diegetic as advanced military technologies. They significantly help the player to aim at targets in VR.
Reload a revolver by flicking hand to open and close its cylinder	*Hot Dogs, Horseshoes & Hand Grenades* (Figure 155)	The interaction builds on the revolver's unique mechanical structure and affordance to achieve realism and efficiency in the reloading process.
Stabilize a rifle with both hands for **aiming**	Standard aiming mechanic in most VR FPS games	The interaction stabilizes the realistic recoil physics and increases the aiming accuracy.
"Third-Person" Scoped Aiming without tunnel	*Pavlov VR* (Figure 136)	An efficient aiming technique used in rifles and sniper rifles to balance the benefit of scoping effect and the visibility of the surroundings

vision		
Three-step manually Reload with snapping and button control optimizations	*Half-Life: Alyx* (Figure 150)	The snapping to insert the clip and the well-mapped button controls for ejecting the clip and reloading are not realistic but achieve realism and increase efficiency in the standard three-step manual reloading process.

Table 8. High-efficiency, low-simulation FPS combat interaction design cases

High-Efficiency, Low-Simulation FPS Combat Interaction Design Cases	References in Case Studies	Description
Aiming assist	*Pistol Whip* (Figure 61)	The interaction achieves contextual efficiency with non-diegetic highlighting visuals and a universal solution that turns any surface climbable.
Body-anchored, tilted hologram map	*Population: ONE* (Figure 164)	The tilted map opens in front of the body does not block the player's sight. It is even close to realism with the hologram.
Grenade with trajectory visuals	*Population: ONE* (Figure 160)	The trajectory visuals and the non-throwing technique increase the efficiency of grenading but lose the realistic physics.
Reload by pointing down or putting back into the holster	*Pistol Whip* (Figure 157), *Sairento VR*	The interactions simplify and automate the reloading process to significantly reduce physical challenges in the manual approach.

Table 9. High-simulation, low-efficiency FPS combat interaction design cases

High-Simulation, Low-Efficiency FPS Combat Interaction Design Cases	References in Case Studies	Description
Grenade by physically triggering and throwing	*Pavlov VR* (Figure 159)	The interaction reproduces the physical pulling to trigger and the throwing behaviors but is not efficient in controlling the throwing and timing.
Mechanical Aiming	*Onward* (Figure 133)	Realistic aiming that requires physical alignment of small aiming slots but relatively less efficient compared with other techniques with assistive visuals and mechanics
Radio and text commands	*Tales of Glory 2 – Retaliation* (Figure 166)	The radio interaction is a good metaphor of using the headset device, but its efficiency is limited in voice interactions, as the text-based commands using pointing controls can be inefficient in FPS contexts.
Tablet map interfaces	*Onward* (Figure 162)	The tablet is a realistic and diegetic interface that can be useful for viewing the map and additional

		operations, but its efficiency is limited by the size and the action of grabbing from a body slot.

Table 10. Low-simulation, low-efficiency FPS combat interaction design cases

Low-Simulation, Low-Efficiency FPS Combat Interaction Design Cases	References in Case Studies	Description
Reload with mechanical structures and constraints	*Population: ONE* (Figure 152), *Pavlov VR* (Figure 153)	The difference between the gun's virtual and real-world use cases and the realistic mechanical structure case leads to a mismatch between the player's reloading behavior to its potential real-world referent, such as reload by using the left hand to pull the right-side handle, causing physical inefficiency or failed attempts.
Bullet traces and fixed kill camera	*Sairento VR* (Figure 138, Figure 139), *Pistol Whip* (Figure 141, Figure 142), *Pavlov VR* (Figure 144), *Onward* (Figure 145)	These interactions may fall into the "poor immersivity" level and cause uncomfortable experiences.

8.2.1.4. RPG Combat Interactions

Regarding RPG combat interactions, the examples from the five categories are towards the optimal space in the framework, as shown in Figure 241. Within each category, readers can find what they think is the best of the best practices; however, I find the variance among them is relatively low. Most of these interactions are based-on gestures and hand manipulations with objects, which themselves have distinct affordances for the interactions to achieve high simulation and efficiency values. Overall, the interactions with head/body movements and non-diegetic visuals and menus tend to have low simulation values, while time becomes a salient factor for efficiency. The realistic weapon swing interactions may lead to uncomfortable experiences and fall into the "poor immersivity" level of low efficiency when there is violent and blood and gore content.

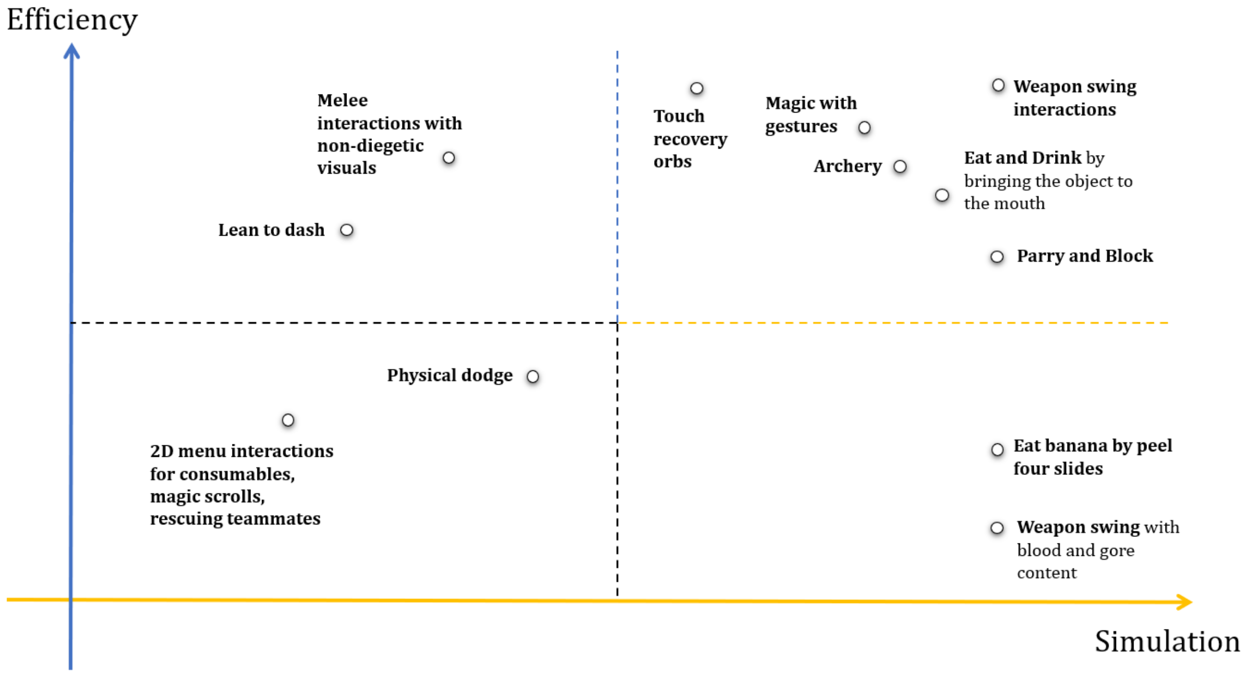


Figure 241. Representative RPG combat interactions: the optimal cases, the high-efficiency-low-simulation cases, the high-simulation-low-efficiency cases, and the low-efficiency-low-simulation cases.

8.2.2. Design Principles and Guidelines

It is worth mentioning that the above best practices and representative examples should not be overly generalized. They are only my account based on the simulation and efficiency frameworks. In the following discussion, I move away from specific examples to offer some higher-level principles and guidelines as a takeaway for designers.

8.2.2.1. Achieving the Optimal of Simulation and Efficiency

In this dissertation, I primarily used the simulation and efficiency framework to map and compare the examples. In this section, I add the analytical lenses and new understandings onto the framework's two value continuums and discuss how the framework can be further used as a design tool.

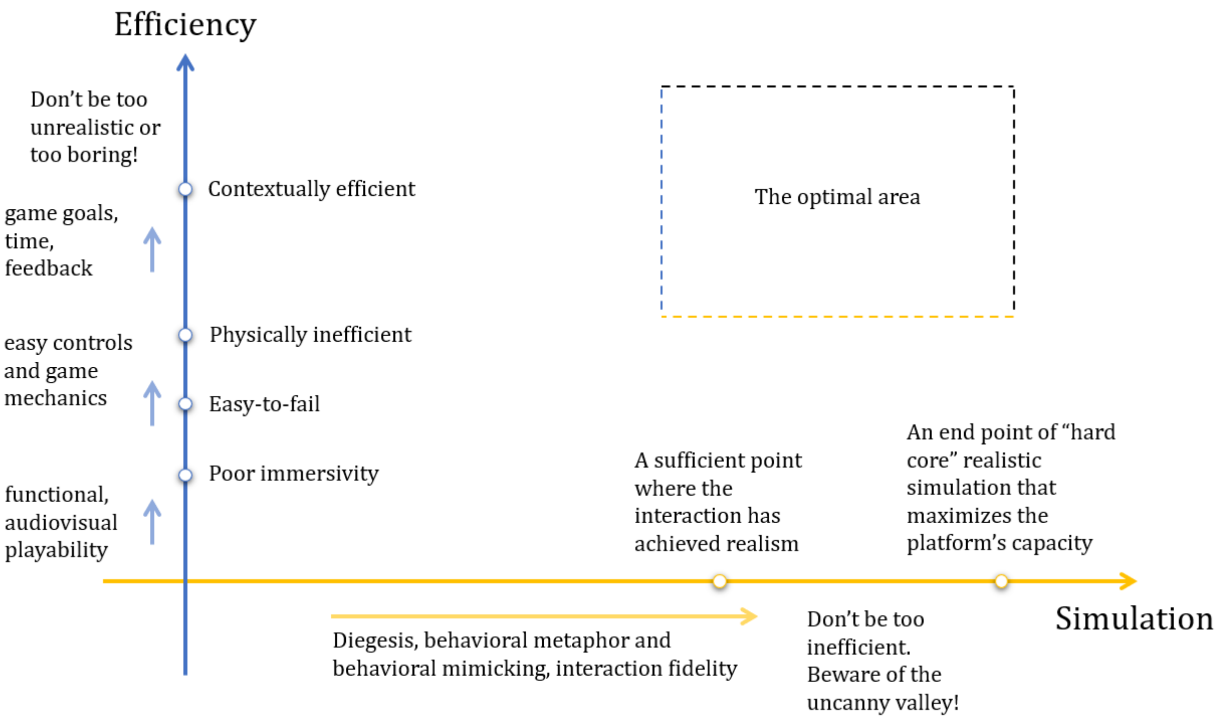


Figure 242. The design process of going towards the optimal area

Since the dissertation views the design of VR game interaction as reconciling the value tensions between simulation and efficiency to eventually reach the optimal area, based on what we have learned so far, I illustrate the process of going along the two value continuums in Figure 242. The following guidelines and principles detail the process.

1. Ensure playability and immersivity. Check the functional and audiovisual components to avoid or mitigate any potentially uncomfortable moments in the experience and keep the immersion as the baseline.

2. Avoid easy-to-fail interactions and physical inefficiency. Implement easy controls and game mechanics to avoid potential failed interaction attempts that deviate players from engaging with the core mechanics and content. To be more inclusive and reach a broader audience in FPS and RPG games, avoid forcing players to perform unhandy physical interactions repeatedly and cater for the seated play mode.

3. Design for contextual efficiency. Further enhance efficiency by examining what efficiency means in your specific gameplay context. For example, use assistive visuals and feedback and reduce the time and steps needed to perform the interaction to achieve the desired game goals.

4. Realism is sufficient. Aiming to achieve realism rather than be overly realistic. Use diegetic interfaces, behavioral metaphor and behavioral mimicking strategies, and high levels of interaction fidelity. Being too realistic will require significantly more effort in overcoming technical challenges and making sure the physics and technical system are free of singularity points or the uncanny valley effect.

5. Balance simulation and efficiency values. Sometimes, the highest values of simulation and efficiency cannot be simultaneously achieved. Being too efficient may lead to unrealistic interactions or boring experiences. Going for the maximized simulation may lead to inefficiency or uncanny valley effects. Therefore, the optimal design requires a joint effort of the above process and a balance between the two values.

8.2.2.2. Consistency with Interaction, Platform, and Game Context

One way to balance simulation and efficiency is to strive for *consistency* with the interaction, the platform, and the game challenge. Consistency is often proposed as a key idea in design and is explicitly mentioned by well-respected usability design principles and heuristics (Nielsen, 1994; Norman, 1999; Shneiderman et al., 2016). In the field of game design, a design model of consistency that particularly inspired this discussion is Scott Nicholson's "Ask Why" model established for creating a better player experience in escape room design[30] (Nicholson, 2016). As I mentioned in my previous case study of escape rooms in chapter two, the model emphasizes the importance to achieve consistency with the forms of the puzzle, the interaction with it, and the meaningful narrative it creates. The same logic can be applied to VR games to design for an interaction that achieves consistency with the affordance of the VR headset and controllers, the visual forms of interfaces, the interaction technique, and the game challenge and narrative within which the interaction is performed.

---

[30] Although the model is originally proposed for escape room design, Nicholson presented it in an analog game study conference GENeration Analog and suggest that it can be used for other game design contexts. I first came across this design model when studying escape room games. I find it is useful to apply to VR games. Both escape room and VR games are immersive interactive experience created by a joint effort of game elements and components within the platform level, the interaction level, and the game context level.

For example, the archery interaction in VR is performed in the embodied way that is based on the hand tracking of the VR motion controllers. Compared with traditional games, VR archery usually takes longer to shoot an arrow and requires more aiming skills to hit the target. Therefore, when the same semantic action is imported to VR games, the other game elements, such as the visual forms of the bow and the game challenge, should be adapted to match the new platform constraints and interaction techniques. For example, in *Sairento VR*'s archery design, the bow can shoot three hyper-realistic arrows at a time to increase the contextual efficiency of hitting targets. In *Shadow Legend VR*, the bow has an aiming slot to increase the efficiency of aligning the two hands and the eye, which is how archery is done in VR. The game challenge of shooting an arrow at the bombs at the feet of a skeleton who does not move and is immune to physical damage but keeps throwing the bombs at the player is a special situation created for using the archery. In *Budget Cuts 2*, the bow features the hyper-reality form that is technologically enhanced with aiming lenses and special crystals as its arrows. The visual form and functions match the robotic theme of the game.

Another example is the aiming slots for guns in FPS games. *Half-Life: Alyx* allows the player to unlock the hologram aiming lens as an upgraded technology to enhance the efficiency with the mechanical aiming slot of the pistol. It provides a larger and semitransparent aiming scope. The hologram also matches the game's setting in the future.

These examples reveal the benefits of having a hyper-realistic game setting or game world. It gives the designer an extra degree-of-freedom to achieve consistency with the interaction components for optimal simulation and efficiency values. The examples in this study's lenses further extend our understandings from previous studies on the effects of hyper-natural components of interaction fidelity on VR locomotion (Nabioyuni & Bowman, 2015) and the design recommendations of using substitutions for physical game challenges and approximation of realism for physical behaviors to suspend disbelief (Rogers et al., 2019). Because in our simulation and efficiency theory, hyper-realistic components are an efficient simulation to achieve realism and reconcile potential value tensions by achieving consistency.

8.3. Future Work and Final Thoughts

The VR game interactions that we have seen in this study are shaping the norms of VR as a gaming platform as it approaches the promised future era of becoming the next interaction platform to connect massive users. At the time of writing this dissertation, technology and gaming industries are hyped about the metaverse[31], which often envisions VR as a key technology and gaming as one of its social applications. Facebook changed its parent company's name to "Meta Platforms" to reflect its vision of being a metaverse company. Online multiplayer games and social platforms such as *Roblox* (Roblox Corporation, 2016), *Minecraft* (Mojang Studios et al., 2011), and *VRChat* (VRChat, 2017) all advertise the concept of metaverse in their own ways. Although tracking down all the narratives of VR and metaverse would be out of the scope, in this final section, I will still focus on the interaction perspective and return to *platform studies* while extending the discussion from current VR games to future applications of VR.

One possible direction of extending this work in the near future is to apply the theoretical and practical outcomes of this study to future VR games and applications. Recent early-access VR games have shown the trend of inheriting and combining some of the best approaches identified in this study. For example, *Hellsweeper VR* is a successor of *Sairento VR* from the developer Mixed Realms. It inherits the teleport-jump, the backflip, the weapon swinging and gun shooting mechanics from *Sairento VR* but also adds the gesture-based magic interactions. *Zenith: The Last City* from TribeVR is a VR MMORPG in its alpha test phase. It implements gliding and climbing to help explore the open world and divides the RPG combat interactions into two classes, the blade master class, which dual-wields swords and the essence mage class, which casts magic spells for ranged attacks or heals. When VR MMORPGs or metaverses become popular, we can expand the set of interactions in this study to include multiplayer and social interactions and apply the value lenses to examine their design. However, I anticipate that the value tensions between simulation and efficiency will be further amplified by the increasing *number of actions* that need to be implemented and performed.

---

[31] At the time of writing this dissertation, the metaverse is still a hyped concept without any commonly acknowledged standards or applications that fully realize its vision. The concept generally refers to a 3D internet of social applications connecting massive users and promotes the utopian narrative of achieving freedom with avatars and virtual environments.

For example, will the VR platform limit the number of abilities and skills in the RPG combat system? What would be the typical number of abilities and skills for each class in a VR MMO, and how would they be cast? With the PC gaming platform, classic MMORPGs such as *World of Warcraft* can easily support over ten or twenty key bindings for various combat abilities for each class. While many VR games may try to replicate the classic RPG design and experience, in actuality they cannot simply achieve that because of the platform limitations, and their game interaction design needs to balance between simulation and efficiency. Will VR players be satisfied with only having several abilities (instead of over ten or twenty) despite they are more embodied and realistic? The gameplay mechanics and content in VR MMOs may also be different from traditional MMOs. For games on mobile phones, such as MOBA games, the number of actions is limited to single digit because of the screen size that can only layout that many of buttons for the right hand to press. MMORPGs on mobile games, as we have seen in the current market, cannot achieve the same scale and experience as they are on the PC platform. As I mentioned in the discussions on consistency, the platform fundamentally changes the interactions, and the interactions further change the design of gameplay mechanics and content, and ultimately, the player experience with the game.

In platform studies, this perspective of the platform being the foundation for its content is reflected in Montfort and Bogost's five-level model of digital media, as shown in Figure 243, where the platform is the base level of all other aspects (Montfort & Bogost, 2009, p. 146). In my previous platform study of escape room games (Jing, 2020), I examined how the room and objects compute logical states and structure player dynamics. The game platform manifests analog computing in a special way by combining the materiality that affords computing (Altice, 2014; Švelch, 2016) and the players' mind and body that performs the act of computing (Bellomy, 2017).

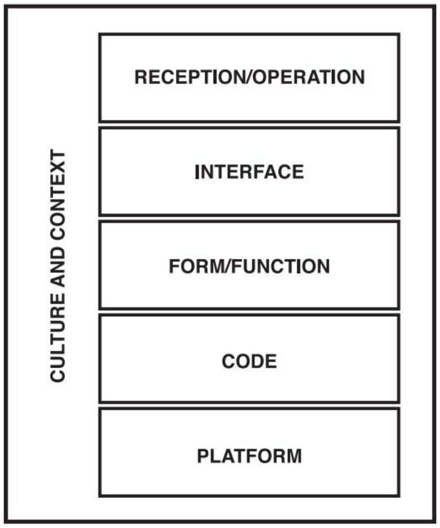


Figure 243. The five levels of digital media, from Montfort and Bogost's *Racing the Beam* (Montfort & Bogost, 2009, p. 146).

Following on the idea of "The medium is the message" (McLuhan, 1964) to reveal the interaction genre that each technical platform is good at supporting, I position this work as a contribution to the platform study of the first-generation commercial VR gaming platforms. Specifically, I have argued that simulation is a salient design value for VR platforms, and for game interaction design, simulation needs to be balanced with efficiency. The case studies show how the VR platform deals with simulation and efficiency in various game interactions.

Despite that the exciting narratives of VR seem to come true with the commercial platforms and content, we should still acknowledge that the limitations of the current VR platform and be critical about how well it may fulfill the visions. In essence, most commercial VR hardware consists of only a headset and two motion controllers that support position and orientation tracking. That's why in some case studies of VR game interactions, their efficiency value is primarily affected by the physical behavior of the lower body. And that is also why we see some existing social VR applications, such as *Microsoft Mesh* and *Altspace*, show avatars without legs (Figure 244). *VRChat*, one of the largest VR social applications, features anime and fictional style avatars. It allows users to upload and tweak their own characters and use extra hardware nodes attached to legs for full-body tracking. Research has identified the gap of full avatar embodiment with the problematic design of "no-legs needed" in video game experience (Hutchison, 2006) and its impact on efficiency, such as in solving team-based puzzles and behaving around obstacles (Pan & Steed, 2019). Because the avatar is central to many social applications and games, I foresee in future platform studies of VR, much remains to be understood about the user/player's acceptance of not having full-body tracking in different application contexts. For example, will users submit to uncanny avatars with rigid bodies or no legs in social VR applications because of other efficiency values? Will players favor the standing full-body tracking mode or seated mode when it comes to reproducing the classic hour-long raids and dungeons in future VR MMOs?

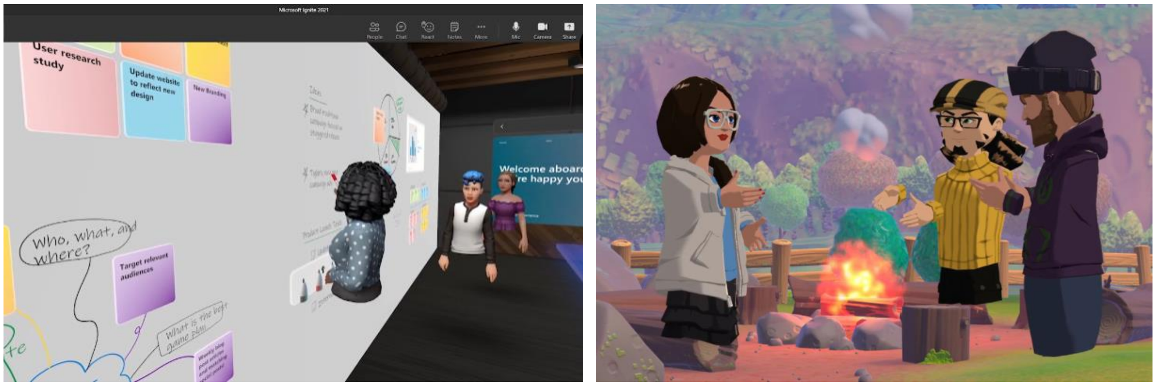


Figure 244. Microsoft Mesh (left) and AltspaceVR (right) only represent the upper bodies. Screenshots from (*Introducing Microsoft Mesh | Here Can Be Anywhere.*, 2021).

Besides the direction of tracking legs and the full-body, the VR platform also evolves with the increasing need to pursue higher realistic simulations with various other control inputs. For example, recent research from Facebook Reality Lab has shown technical progress on representing natural articulated hand models (Zhang et al., 2021), tracking facial expressions (Bikumandla & Han, 2021), and estimating the full-body posture (Jiang & Ithapu, 2021). Wearable, untethered hand tracking with passing magnets is another alternative to vision-based tracking (Chen et al., 2021). These are promising features for the next generations of VR platforms. While the current optimal area of simulation and efficiency tends to involve hand-based interactions, it is interesting to see how the new features will further update and complexify simulation and efficiency values in the interaction design. It is worth noting that the hand-tracking gesture input will replace the motion controller, while others such as the facial expression, posture, and even brain-computer interfaces will be added to co-function with the controller input. I expect that it will be more challenging for gesture-based VR applications without the motion controllers to reconcile the value tensions within locomotion interactions. Facial expressions and full-body posture will potentially extend the current shortage of emotional expressions and non-verbal communication in VR, as identified by (Tanenbaum et al., 2020).

Each of these promising input features manifests as a modulized micro-platform when integrated into VR because each of them generates computational states that abstract its respective physical attributes in their unique ways. Therefore, from the platform perspective, each interaction design has unique considerations. They may also be subjective to the simulation and efficiency framework and need to be designed in consistency with the application context. For example, imagine facial and posture tracking as interaction inputs. How will users control to express their intentions? Will they be always showing their *real* facial expressions and body posture at all times? What will an interface be like, if any, to mediate these non-verbal communications? Can users customize their simulated smiles?

Furthermore, consider brain-computer interfaces (BCIs) for VR. The same above questions still hold as design challenges. BCIs typically use algorithms to extract meaningful metrics such as attention and meditation values and motor imageries from EEG data. However, these metrics probably do not precisely represent the user's true mental states and intentions, despite being translated into numeric values. Therefore, the BCI platform may lead to design space with what I like to call *fuzzy interactions*: users may not be able to execute the controls immediately and may feel uncertain about whether the feedback they see matches with their intended controls. Still, research suggests that BCI can be used in VR for object selection, object manipulation, navigation, and application control (Lotte et al., 2013). And an ultimate vision of VR is that it is entirely based on BCI, like what *The Matrix, Avatar,* and *Sword Art Online* have shown to the public. Nevertheless, I remain skeptical about the usage of BCI for any control tasks in VR. I think it will be less efficient than traditional hardware controllers because of its accuracy. It needs to deal with issues such as confirming the intended thought and avoiding unintended random thoughts.

Besides controlling, I think BCI has other potential uses for activities such as meditation and emotional add-ons.

The detailed exploration of every new module that can be integrated into VR is definitely out of the scope for this section. Here, I suggest future work to examine the platform and its computing mechanism to understand what kinds of interactions are best suited when those promised VR technologies come out. The value lenses that I theorized in this work may still apply in future platform studies of VR with every new technology added to the platform.

It is also worth noting that culture and context play an important role in the emergence of new platforms. They set the visions and concepts, tell stories and promises for people to accept new technologies and continuously work to develop them. Hiroshi Ishii, one of the most influential scholars in HCI, argued that vision-based design has a longer lifespan of decades compared with shorter-term need-based and technology-based design (Ishii et al., 2012). VR is no exception - from the "ultimate display" concept (Sutherland, 1965) to today's metaverse, VR has constantly been viewed as a vision rather than a specific technology. Films and anime have clearly delivered metaphors of VR, but the technology industry itself has always contributed a lot to VR discourse in history.

Chris Chesher surveyed the emerging discourses of VR in the technology industry in the 1990s and the cultural context into which VR was introduced at that time (Chesher, 1994). He argued that the process of VR coming to acceptance involves more than creating better hardware and software, but also its marketing and promotion to better suit mainstream values. He showed narratives of VR, such as "VR is the inevitable and logical conclusion to a historical process", "absolutely everything can be and should be represented in digital form", "the paradigm shift of computers being reality generators beyond symbol processors", and "VR is a new world created by elites in Silicon Valley to escape the reality that is difficult to change," etc. At that time, many companies and organisations attempted to market VR with various prototypes. Clearly, many have failed or faded out in the past thirty years. Despite the failure in the past, today's new major companies in VR, including those who manufacture VR devices, develop 3D engines, and create VR games, have continued investing in pursuing the technological utopian, releasing more concrete products and updated metaphors, such as the metaverse.

Before continuing to discuss VR and metaverse as a technological utopian in the cultural context, to put them in a simple word for all these past and present narratives of VR, I think VR is a simulation (in the spirit of this work). And my final thoughts on simulation (and efficiency) are with respect to perhaps the most profound work on the concept of simulation - the philosophical treatise *Simulacra and Simulation* by Jean Baudrillard (Baudrillard, 1994). The penetration of commercial VR into our everyday life and culture, especially the narrative of escaping the real world and living in VR and metaverses as simulations of reality, would be what Baudrillard claims as the *precession of simulacra* (p. 1), which replaces the meanings in reality with symbols and signs that do not have originals in reality. He draws a beautiful analogy of simulation from a fable, in which a map exactly covers all the details of a territory. When the territory falls apart, it no longer precedes the map. Instead, in the end, it is the map that precedes the territory and accommodates its people, saturating their experience of reality with images from the map maker. Baudrillard delineates four successive phases of an image deviating from the reality:

> "1) it is the reflection of a profound reality;
> 2) it masks and denatures a profound reality;
> 3) it masks the *absence* of a profound reality;
> 4) it has no relation to any reality whatsoever; it is its own pure simulacrum." (Baudrillard, 1994, p. 6)

Indeed, the VR examples we have seen in this study have forms that fall into these stages. Outsiders of VR may view a VR game and its elements as simulacra, which are purely for the player's ludic pleasure and do not matter or have original meanings in the real world. For example, the interactions of swinging hands to hit rhythmic blocks with direction signs, pumping big-number damage on hyper-realistic enemies, and following non-diegetic icons all involve forms that are no longer meaningful in reality. However, the player may still experience meaningful play and preserve that the interactions are meaningful and realistic in their nature.

In fact, one of the core pleasures of VR is to experience not only different realities but also various kinds of hyper-reality through simulation. While Baudrillard tends to critique the hyper-real from a nihilism perspective, I am more optimistic. But first, I want to acknowledge VR's potential cultural and societal impact as we may indulge ourselves in such a platform for simulation.

Baudrillard was concerned about the disappearance of original meanings due to the industrial revolutions. In a simulation society – like the one we live in - so many things and even people that take our attention in everyday life become digital symbols and simulations, and their analog representations and labor are lost and forgotten. For example, during the Covid-19 pandemic, we worked and interacted with people represented in 2D profile photos on social media and digital workspace. With VR and metaverses, we may eventually work and interact with people represented in 3D avatars every day, as promoted by concept videos. But their symbolic nature is the same.

The future cutting-edge VR platform for consumers to live in a simulation will still, in its nature, be a crystallization of invisible labor within capitalism. Like what game scholar Aaron Trammell suggests for platform studies, the production of a game platform is an assemblage of work that involves a variety of creative labor practices from both professional developers and fan communities (Trammell, 2019). With VR, future platform studies can document the labor behind making the platform to preserve history and remind people of the “territory” outside the “map”. Furthermore, even within the map, new histories will be made. As Chris Chesher suggested in his study of the colonization of VR, the invisible hand of economics will still divide users into different social classes in VR as they are out of VR (Chesher, 1994). In this regard, efficient features in VR and metaverses may one day even become products for sale, as efficiency is always a value in capitalism.

Still, my optimism towards future VR and metaverses has several reasons. Firstly, I am not too worried about the potential negative cultural, societal, and historical implications of VR expediting a simulation society. In history, every new medium, from audio to the internet to the metaverse, has been criticized in similar ways. It is not technology that fundamentally changes society, but the values they promote and are accepted as the mainstream that evolve human civilization. The VR platform promotes simulation and efficiency values. The hyper-realistic interactions are suitable for VR, such as teleportation and other hyper-realistic and efficient design. VR is defined by hyper-reality and makes hyper-reality feel real. My simulation and efficiency framework highlights optimal interaction design based on hyper-realistic forms and techniques, which enhance the efficiency of our real-world interactions while still achieving realism. The visions of metaverses, such as delivered from promotion videos, seem to tell people that movement is not a problem in VR because they can teleport, they can do their routines of work more efficiently, and they can experience more things and faster. The only things they need to do outside VR are eating and sleeping for their physical sustainability. Although this sounds crazy and dystopian, it is close to what many of us have already been doing during the Covid-19 pandemic. We sleep, order food delivery[32], and meet and work virtually. The VR platform will just massively enrich the content if we choose to continue living in this way. In this sense, we can better understand virtual reality as *extended* reality. It extends our efficiency in performing meaningful actions beyond the limitations of our physical strength. It extends our view of the world and our imaginations beyond the physical forms. For VR to be more successful and appealing to massive users in the future, I suggest that a feedback loop between VR and the real world must be achieved for VR to be a true extension of reality - one that brings substantial benefit to our career, business, education, and social relationships. Such is what people may call a metaverse, or a 3D internet, where games can be an appetizer or a pivotal application to bring people into the virtual world.

Secondly, I am not skeptical that the metaverse is just a revision of the same technologies that we already have (including those that have "failed" in the past). Many have argued that there is nothing new in the metaverse and that it is just an exciting concept. Ironically, those arguments themselves are "new" because of the concept of metaverse, and they may even motivate the builders who do creative labor. As I have drawn from (Chesher, 1994), the process of introducing a new technology has always

---

[32] In the science fiction *Snow Crash* (Stephenson, 2003), where the term metaverse was invented, Neal Stephenson described the pizza delivery uses extremely high-technology, geared, fast cars to ensure the food is delivered on time, but still faces potential robbery of the car.

involved with its cultural acceptance. The concept of VR has been guiding technology development over the past decades, and it will undoubtedly continue in the years to come. We just have to be patient. As someone who self-considers being a builder in this area, I am optimistic because I have witnessed the technology growing from the 90s to the current state. In fact, many new technology inventions are reassembling the same technologies but in new ways to create new experiences. And at certain times, as they become prevalent, people name them as new technologies. When writing this section, several game studios have released their plans for expanding their IP to align with the metaverse visions, including building larger open worlds and providing affordances for forming social communities within the game and user-generated content. As players, it is exciting to see video games evolving and providing us with new content and experiences, thanks to the new concept.

Finally, I envision there will be a 3D *Youtube* or *Tiktok* in the future of metaverse. These video streaming and sharing platforms have revolutionized the Internet, and the same historical patterns and rules of user-generated content may reoccur in the metaverse. However, in the current landscape of VR environments, such as those created in *VRChat*, many are static scenes without meaningful things to do and people who play, socialize, or make a living within the space. This dissertation work has examined how a set of commonly used “verbs” are translated into interactions on commercial VR platforms. As VR technologies continue to extend our reality and become more accessible in the future, the fundamental design values of simulation and efficiency will continue to guide interaction design and remain relevant in creative practices.

GAME BIBLIOGRAPHY

Due to the nature of this work reviewing a considerable number of VR games, I added this separate game bibliography for the ease of reference.

Aldin Dynamics. (2021). *Waltz of the Wizard: Natural Magic*. Aldin Dynamics.

Bethesda Game Studios. (2017). *The Elder Scrolls V: Skyrim VR*. Bethesda Softworks.

BigBox VR, Inc. (2020). *Population: ONE*. BigBox VR, Inc.

BlackTale Games. (2020). *Tales of Glory 2 – Retaliation*. BlackTale Games.

Blizzard Entertainment. (1998). *Diablo*. Blizzard Entertainment.

Blizzard Entertainment. (2004). *World of Warcraft*. Blizzard Entertainment.

Crytek. (2021). *The Climb 2*. Crytek.

Deep Type Games. (2019). *Hellsplit: Arena*. Deep Type Games.

Downpour Interactive. (2016). *Onward*. Downpour Interactive, Coatsink.

E McNeill. (2020). *Ironlights*. E McNeill.

Free Lives, & 24 Bit Games Pty. Limited. (2017). *GORN*. Devolver Digital.

Greensky Games, Inc. (2020). *SwarmVR*. Greensky Games, Inc.

inXile entertainment. (2020). *Frostpoint VR: Proving Grounds*. Thirdverse.

Joy Way. (2020). *STRIDE*. Joy Way.

Mixed Realms, & Swag Soft Holdings Pte Ltd. (2016). *Sairento VR*. Mixed Realms, Perpetual Europe.

Mojang Studios, Other Ocean Interactive, 4J Studios, & Xbox Game Studios. (2011). *Minecraft*.

Neat Corporation, & Fast Travel Games. (2019). *Budget Cuts 2: Mission Insolvency*. Neat

Corporation.

Ready At Dawn. (2017). *Echo VR*. Oculus Studios.

Roblox Corporation. (2016). *Roblox*.

RUST LTD. (2016). *Hot Dogs, Horseshoes & Hand Grenades*. RUST LTD.

Schell Games. (2020). *Until You Fall*. Schell Games.

Schell Games. (2021). *I Expect You To Die 2*. Schell Games.

Stress Level Zero. (2019). *Boneworks*. Stress Level Zero.

Survios. (2021). *The Walk Dead Onslaught*. Survios.

Thirdverse. (2019). *SWORDS of GARGANTUA*. Thirdverse.

TiMi Studio Group. (2015). *Honor of Kings*. Tencent Games.

Urban Wolf Games Inc. (2021). *Legendary Tales*. Urban Wolf Games Inc.

Valve Corporation. (2020). *Half-Life: Alyx*. Valve Corporation.

Vankrupt Games. (2017). *Pavlov VR*. Vankrupt Games.

VitruviusVR. (2019). *Shadow Legend VR*. VitruviusVR.

VRChat. (2017). *VRChat*.

WarpFrog. (2018). *Blade and Sorcery*. WarpFrog.

BIBLIOGRAPHY

Al Zayer, M., MacNeilage, P., & Folmer, E. (2020). Virtual Locomotion: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, *26*(6), 2315–2334. https://doi.org/10.1109/TVCG.2018.2887379

Aldin Dynamics. (2021). *Waltz of the Wizard: Natural Magic*. Aldin Dynamics.

Alexander, C. (1977). *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press.

Altice, N. (2014, November 3). The Playing Card Platform. *Analog Game Studies*. http://analoggamestudies.org/2014/11/the-playing-card-platform/

Altice, N. (2015). *I am error: The Nintendo family computer/entertainment system platform*. The MIT Press.

Ang, I. (1995). The nature of the audience. In *Questioning the media: A critical introduction* (Second Edition, pp. 207–220). Thousand Oaks, CA: Sage.

Apperley, T., & Parikka, J. (2018). Platform Studies' Epistemic Threshold. *Games and Culture*, *13*(4), 349–369. https://doi.org/10.1177/1555412015616509

Arsenault, D. (2008). Guitar Hero: "Not Like Playing Guitar At All"? *Loading...*, *2*(2), Article 2. https://loading.journals.publicknowledgeproject.org/loading/index.php/loading/article/view/32

Banissy, M. J., & Ward, J. (2007). Mirror-touch synesthesia is linked with empathy. *Nature Neuroscience*, *10*(7), 815–816. https://doi.org/10.1038/nn1926

Bardram, J. E., & Bertelsen, O. W. (1995). Supporting the development of transparent interaction. In *Human-Computer Interaction* (Vol. 1015/1995, pp. 79–90). Springer Berlin / Heidelberg.

Barthes, R. (1977). *Image-music-text*. Macmillan.

Baudrillard, J. (1994). *Simulacra and simulation*. University of Michigan press.

Bellomy, I. (2017, March 20). What Counts: Configuring the Human in Platform Studies. *Analog Game Studies*. http://analoggamestudies.org/2017/03/what-counts/

Bethesda Game Studios. (2017). *The Elder Scrolls V: Skyrim VR*. Bethesda Softworks.

Bhargava, A., Bertrand, J. W., Gramopadhye, A. K., Madathil, K. C., & Babu, S. V. (2018). Evaluating Multiple Levels of an Interaction Fidelity Continuum on Performance and Learning in Near-Field Training Simulations. *IEEE Transactions on Visualization and*

Computer Graphics, *24*(4), 1418–1427. https://doi.org/10.1109/TVCG.2018.2794639

BigBox VR, Inc. (2020). *Population: ONE*. BigBox VR, Inc.

Bikumandla, M., & Han, C. (2021). *Systems and methods for facial expression tracking* (United States Patent No. US20210223856A1). https://patents.google.com/patent/US20210223856A1/en

Bizzocchi, J. (2003). *Ceremony of Innocence and the Subversion of Interface: Cursor Transformation as a Narrative Device*. Digital Arts And Culture (DAC), Melbourne. http://hypertext.rmit.edu.au/dac/papers/Bizzocchi.pdf

Bizzocchi, J., Lin, B., & Tanenbaum, T. J. (2011). Games, Narrative, and the Design of Interface. *International Journal of Art and Technology (IJART)*, *4*(4), 460–479.

Bizzocchi, J., & Tanenbaum, T. J. (2011). *Well read: Applying close reading techniques to gameplay experiences* (D. Davidson, Ed.; pp. 262–290). ETC Press. http://dl.acm.org/citation.cfm?id=2031858.2031881

Bjork, S., & Holopainen, J. (2004). *Patterns in game design (game development series)*. http://www.citeulike.org/group/1820/article/423551

BlackTale Games. (2020). *Tales of Glory 2 – Retaliation*. BlackTale Games.

Blaga, A. D., Frutos-Pascual, M., Creed, C., & Williams, I. (2020). Too Hot to Handle: An Evaluation of the Effect of Thermal Visual Representation on User Grasping Interaction in Virtual Reality. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–16. https://doi.org/10.1145/3313831.3376554

Blizzard Entertainment. (1998). *Diablo*. Blizzard Entertainment.

Blizzard Entertainment. (2004). *World of Warcraft*. Blizzard Entertainment.

Bogart, A., & Landau, T. (2005). *The Viewpoints Book: A Practical Guide to Viewpoints and Composition*. Theatre Communications Group.

Bogost, I. (2006). *Unit Operations: An Approach to Videogame Criticism*. MIT Press.

Bogost, I. (2007). *Persuasive Games: The Expressive Power of Videogames*. MIT Press.

Boletsis, C. (2017). The New Era of Virtual Reality Locomotion: A Systematic Literature Review of Techniques and a Proposed Typology. *Multimodal Technologies and Interaction*, *1*(4), 24. https://doi.org/10.3390/mti1040024

Boletsis, C., & Cedergren, J. E. (2019). VR Locomotion in the New Era of Virtual Reality: An Empirical Comparison of Prevalent Techniques. *Advances in Human-Computer Interaction*, *2019*, e7420781. https://doi.org/10.1155/2019/7420781

Bolter, J. D., & Grusin, R. (1999). Immediacy, Hypermediacy, and Remediation. In *Remediation: Understanding New Media* (pp. 20–50). The MIT PRess.

Bowman, D. A., McMahan, R. P., & Ragan, E. D. (2012). Questioning naturalism in 3D user interfaces. *Communications of the ACM*, *55*(9), 78–88. https://doi.org/10.1145/2330667.2330687

Bozgeyikli, E., Raij, A., Katkoori, S., & Dubey, R. (2016). Point & Teleport Locomotion Technique for Virtual Reality. *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, 205–216. https://doi.org/10.1145/2967934.2968105

Brewster, S. A., & Brown, L. M. (2004). *Tactons: Structured tactile messages for non-visual information display*.

Brooks, F. P. (1999). What's real about virtual reality? *IEEE Computer Graphics and Applications*, *19*(6), 16–27. https://doi.org/10.1109/38.799723

Caroux, L., Isbister, K., Le Bigot, L., & Vibert, N. (2015). Player–video game interaction: A systematic review of current concepts. *Computers in Human Behavior*, *48*, 366–381. https://doi.org/10.1016/j.chb.2015.01.066

Chen, D., Wang, M., He, C., Luo, Q., Iravantchi, Y., Sample, A., Shin, K. G., & Wang, X. (2021). MagX: Wearable, untethered hands tracking with passive magnets. *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 269–282. https://doi.org/10.1145/3447993.3483260

Chesher, C. (1994). Colonizing virtual reality: Construction of the discourse of virtual reality. *Cultronix*, *1*(1), 1–27.

Crawford, C. (1984). *The art of computer game design*.

Creswell, J. W. (2003). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Sage Publications.

Crogan, P. (2011). *Gameplay mode: War, simulation, and technoculture* (Vol. 36). U of Minnesota Press.

Crogan, P., & Aarseth, E. (2008). Games, simulations and serious fun: An interview with Espen Arseth. *Scan: J. Media Arts Culture*, *5*(1).

Crytek. (2021). *The Climb 2*. Crytek.

Csikszentmihalyi, M. (1990). *Flow: The Psychology of Optimal Experience*. Harper Perennial.

Danyluk, K., Ens, B., Jenny, B., & Willett, W. (2021). A Design Space Exploration of Worlds in Miniature. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–15. https://doi.org/10.1145/3411764.3445098

de Smale, S., Kors, M. J. L., & Sandovar, A. M. (2019). The Case of This War of Mine: A Production Studies Perspective on Moral Game Design. *Games and Culture*, *14*(4), 387–409. https://doi.org/10.1177/1555412017725996

Deep Type Games. (2019). *Hellsplit: Arena*. Deep Type Games.

Di Luca, M., Seifi, H., Egan, S., & Gonzalez-Franco, M. (2021). Locomotion Vault: The Extra Mile in Analyzing VR Locomotion Techniques. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (pp. 1–10). Association for Computing Machinery. https://doi.org/10.1145/3411764.3445319

Dormans, J. (2011). Beyond Iconic Simulation. *Simulation & Gaming*, *42*(5), 610–631. https://doi.org/10.1177/1046878111426963

Dourish, P. (2004). What we talk about when we talk about context. *Personal and Ubiquitous Computing*, *8*(1), 19–30.

Downpour Interactive. (2016). *Onward*. Downpour Interactive, Coatsink.

E McNeill. (2020). *Ironlights*. E McNeill.

Egliston, B., & Carter, M. (2020). Oculus imaginaries: The promises and perils of Facebook's virtual reality. *New Media & Society*, 1461444820960411. https://doi.org/10.1177/1461444820960411

Ermi, L., & Mäyrä, F. (2005). Fundamental components of the gameplay experience: Analysing immersion. *Worlds in Play: International Perspectives on Digital Games Research*, 37. https://books.google.com/books?hl=en&lr=&id=WykINIyYSb0C&oi=fnd&pg=PA37&dq=Fundamental+components+of+the+gameplay+experience:+Analysing+immersion&ots=w5ONRIO8Y0&sig=9QANrZ2K-ZCp8rY-H5RkY-h_1rU#v=onepage&q&f=false

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, *47*(6), 381–391. https://doi.org/10.1037/h0055392

Flanagan, M. (2009). *Critical play: Radical game design*. MIT press.

Flanagan, M., Howe, D. C., & Nissenbaum, H. (2005). Values at play: Design tradeoffs in socially-oriented game design. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 751–760. https://doi.org/10.1145/1054972.1055076

Flanagan, M., & Nissenbaum, H. (2014). *Values at play in digital games*. MIT Press.

Free Lives, & 24 Bit Games Pty. Limited. (2017). *GORN*. Devolver Digital.

Friedman, B. (1996). Value-sensitive design. *Interactions*, *3*(6), 16–23.

Friedman, B., Kahn, P. H., & Borning, A. (2008). Value sensitive design and information systems. *The Handbook of Information and Computer Ethics*, 69–101.

Frøkjær, E., Hertzum, M., & Hornbæk, K. (2000). Measuring usability: Are effectiveness, efficiency, and satisfaction really correlated? *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 345–352. https://doi.org/10.1145/332040.332455

Frommel, J., Sonntag, S., & Weber, M. (2017). Effects of controller-based locomotion on player experience in a virtual reality exploration game. *Proceedings of the 12th International Conference on the Foundations of Digital Games*, 1–6. https://doi.org/10.1145/3102071.3102082

Gamma, E. (1995). *Design patterns: Elements of reusable object-oriented software.* Pearson Education India.

Gazzard, A. (2016). *Now the chips are down: The BBC micro*. MIT Press.

Grand View Research. (2021). *Virtual Reality Market Share & Trends Report, 2021-2028*. https://www.grandviewresearch.com/industry-analysis/virtual-reality-vr-market

Greensky Games, Inc. (2020). *SwarmVR*. Greensky Games, Inc.

Holm, I. (2006). *Ideas and Beliefs in Architecture and Industrial design: How attitudes, orientations, and underlying assumptions shape the built environment* (Vol. 22). Ivar Holm.

Huizinga, J. (1955). *Homo Ludens: A Study of the Play-Element in Culture*. The Beacon Press.

Hunicke, R., Leblanc, M., & Zubek, R. (2004). MDA: A formal approach to game design and game research. *In Proceedings of the Challenges in Games AI Workshop, Nineteenth National Conference of Artificial Intelligence*, 1–5.

Hutchison, A. (2006). Where are my legs? Embodiment Gaps in Avatars. *Proceedings of the 2006 International Conference on Game Research and Development*, 104–111.

International Organization for Standardization. (1998). *ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs): Part 11: Guidance on usability*.

Introducing Microsoft Mesh | Here can be anywhere. (2021). https://www.microsoft.com/en-us/mesh

inXile entertainment. (2020). *Frostpoint VR: Proving Grounds*. Thirdverse.

Ishii, H., Lakatos, D., Bonanni, L., & Labrune, J.-B. (2012). Radical atoms: Beyond tangible bits, toward transformable materials. *Interactions*, *19*(1), 38–51.

Järvinen, A., Heliö, S., & Mäyrä, F. (2002). Communication and community in digital entertainment services. *University of Tampere: Hypermedia Laboratory Net Series*, *2*, 9–43.

Jerald, J. (2015). *The VR Book: Human-Centered Design for Virtual Reality*. Association for Computing Machinery and Morgan & Claypool.

Jiang, H., & Ithapu, V. K. (2021). Egocentric Pose Estimation from Human Vision Span. *ArXiv:2104.05167 [Cs]*. http://arxiv.org/abs/2104.05167

Jing, K. (2020). Computing Models of Escape Rooms. *Analog Game Studies*, *VII*(II). https://analoggamestudies.org/2020/12/computing-models-of-escape-rooms/

Jing, K., Nygaard, N., & Tanenbaum, T. J. (2017). Magia Transformo: Designing for Mixed Reality Transformative Play. *Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play*, 421–429. https://doi.org/10.1145/3130859.3131339

Jones, S. E., & Thiruvathukal, G. K. (2012). *Codename revolution: The Nintendo Wii platform*. MIT Press.

Joy Way. (2020). *STRIDE*. Joy Way.

Juul, J. (2005). *Half-Real: Video Games between Real Rules and Fictional Worlds*. MIT Press.

Kreimeier, B. (2002). The case for game design patterns. *Gamasutra*. http://www.gamasutra.com/view/feature/132649/the_case_for_game_design_patterns.php

Krompiec, P., & Park, K. (2019). Enhanced Player Interaction Using Motion Controllers for First-Person Shooting Games in Virtual Reality. *IEEE Access*, *7*, 124548–124557. https://doi.org/10.1109/ACCESS.2019.2937937

Kurzweil, R. (2005). *The singularity is near: When humans transcend biology*. Penguin.

Lankoski, P., & Björk, S. (2015). *Game Research Methods* (P. Lankoski & S. Björk, Eds.; pp. 23–35). ETC Press. http://dl.acm.org/citation.cfm?id=2812774.2812779

Limperos, A. M., Schmierbach, M. G., Kegerise, A. D., & Dardis, F. E. (2011). Gaming Across Different Consoles: Exploring the Influence of Control Scheme on Game-Player Enjoyment. *Cyberpsychology, Behavior, and Social Networking*, *14*(6), 345–350. https://doi.org/10.1089/cyber.2010.0146

Lin, B. (2007). *Narrative interface design: The use of interface elements to enhance the narrative experience in videogames* [Thesis, School of Interactive Arts & Technology - Simon Fraser University]. https://summit.sfu.ca/item/8286

Lotte, F., Faller, J., Guger, C., Renard, Y., Pfurtscheller, G., Lécuyer, A., & Leeb, R. (2013). Combining BCI with Virtual Reality: Towards New Applications and Improved BCI. In B. Z. Allison, S. Dunne, R. Leeb, J. Del R. Millán, & A. Nijholt (Eds.), *Towards Practical Brain-Computer Interfaces: Bridging the Gap from Research to Real-World Applications* (pp. 197–220). Springer. https://doi.org/10.1007/978-3-642-29746-5_10

Macklin, C., & Sharp, J. (2016). *Games, Design and Play: A detailed approach to iterative game design*. Addison-Wesley Professional.

Martel, E., & Muldner, K. (2017). Controlling VR games: Control schemes and the player experience. *Entertainment Computing*, *21*, 19–31. https://doi.org/10.1016/j.entcom.2017.04.004

Mäyrä, F., & Ermi, L. (2011). Fundamental components of the gameplay experience. *DIGAREC Series*, *6*, 88–115.

McLuhan, M. (1964). *Understanding Media: The Extensions of Man* (W. T. Gordon, Ed.). McGraw-Hill.

McMahan, R. P. (2011). *Exploring the Effects of Higher-Fidelity Display and Interaction for Virtual Reality Games* [Virginia Tech]. https://vtechworks.lib.vt.edu/handle/10919/30123

McMahan, R. P., Alon, A. J. D., Lazem, S., Beaton, R. J., Machaj, D., Schaefer, M., Silva, M. G., Leal, A., Hagan, R., & Bowman, D. A. (2010). Evaluating natural interaction techniques in video games. *2010 IEEE Symposium on 3D User Interfaces (3DUI)*, 11–14. https://doi.org/10.1109/3DUI.2010.5444727

McMahan, R. P., Bowman, D. A., Zielinski, D. J., & Brady, R. B. (2012). Evaluating Display Fidelity and Interaction Fidelity in a Virtual Reality Game. *IEEE Transactions on Visualization and Computer Graphics*, *18*(4), 626–633. https://doi.org/10.1109/TVCG.2012.43

McMahan, R. P., Lai, C., & Pal, S. K. (2016). Interaction Fidelity: The Uncanny Valley of Virtual Reality Interactions. In S. Lackey & R. Shumaker (Eds.), *Virtual, Augmented and Mixed Reality* (pp. 59–70). Springer International Publishing. https://doi.org/10.1007/978-3-319-39907-2_6

Miller, J. K., Friedman, B., Jancke, G., & Gill, B. (2007). Value tensions in design: The value sensitive design, development, and appropriation of a corporation's groupware system. *Proceedings of the 2007 International ACM Conference on Supporting Group Work*, 281–290. https://doi.org/10.1145/1316624.1316668

Mine, M. R., Brooks, F. P., & Sequin, C. H. (1997). Moving objects in space: Exploiting proprioception in virtual-environment interaction. *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, 19–26. https://doi.org/10.1145/258734.258747

Mirror-touch synesthesia. (2021). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Mirror-touch_synesthesia&oldid=1044641745

Mixed Realms, & Swag Soft Holdings Pte Ltd. (2016). *Sairento VR*. Mixed Realms, Perpetual Europe.

Mojang Studios, Other Ocean Interactive, 4J Studios, & Xbox Game Studios. (2011). *Minecraft*.

Montfort, N., & Bogost, I. (2009). *Racing the beam: The Atari video computer system*. Mit Press.

Mori, M., MacDorman, K. F., & Kageki, N. (2012). The uncanny valley [from the field]. *IEEE Robotics & Automation Magazine*, *19*(2), 98–100.

Nabioyuni, M., & Bowman, D. A. (2015). An evaluation of the effects of hyper-natural components of interaction fidelity on locomotion performance in virtual reality. *Proceedings of the 25th International Conference on Artificial Reality and Telexistence and 20th Eurographics Symposium on Virtual Environments*, 167–174.

Nabiyouni, M., Saktheeswaran, A., Bowman, D. A., & Karanth, A. (2015). Comparing the performance of natural, semi-natural, and non-natural locomotion techniques in virtual reality. *2015 IEEE Symposium on 3D User Interfaces (3DUI)*, 3–10. https://doi.org/10.1109/3DUI.2015.7131717

Neat Corporation, & Fast Travel Games. (2019). *Budget Cuts 2: Mission Insolvency*. Neat Corporation.

Ng, A. K. T., Chan, L. K. Y., & Lau, H. Y. K. (2020). A study of cybersickness and sensory conflict theory using a motion-coupled virtual reality system. *Displays*, *61*, 101922. https://doi.org/10.1016/j.displa.2019.08.004

Nicholson, S. (2015). Peeking behind the locked door: A survey of escape room facilities. *White Paper Available at Http://Scott Nicholson. Com/Pubs/Erfacwhite. Pdf*. http://scottnicholson.com/pubs/erfacwhite.pdf

Nicholson, S. (2016). Ask why: Creating a better player experience through environmental storytelling and consistency in escape room design. *Proc. Int. Acad. Conf. Meaningful Play*, 521–556.

Nielsen, J. (1994). *10 Usability Heuristics for User Interface Design*. Nielsen Norman Group. https://www.nngroup.com/articles/ten-usability-heuristics/

Norman, D. A. (1988). *The Design of Everyday Things*. Doubelday/Currency.

Norman, D. A. (1999). Affordance, Conventions, and Design. *Interactions*, *6*(3), 38–43. https://doi.org/10.1145/301153.301168

Norman, D. A. (2008). THE WAY I SEE ITSignifiers, not affordances. *Interactions*, *15*(6), 18–19. https://doi.org/10.1145/1409040.1409044

Pal, S. K., Khan, M., & McMahan, R. P. (2016). The benefits of rotational head tracking. *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, 31–38.

Pan, Y., & Steed, A. (2019). How Foot Tracking Matters: The Impact of an Animated Self-Avatar on Interaction, Embodiment and Presence in Shared Virtual Environments. *Frontiers in Robotics and AI*, *6*, 104. https://doi.org/10.3389/frobt.2019.00104

Parker, J. R., & Becker, K. (2013). The Simulation-Game Controversy: What is a Ludic Simulation? *International Journal of Gaming and Computer-Mediated Simulations*, *5*(1), 1–12. https://doi.org/10.4018/jgcms.2013010101

Pausch, R., Proffitt, D., & Williams, G. (1997). Quantifying immersion in virtual reality. *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, 13–18.

Pearce, S. M. (2012). Objects as meaning; or narrating the past. In *Interpreting objects and collections* (pp. 31–41). Routledge.

Ready At Dawn. (2017). *Echo VR*. Oculus Studios.

Reason, J. T. (1978). Motion sickness adaptation: A neural mismatch model. *Journal of the Royal Society of Medicine*, *71*(11), 819–829.

Research and Markets. (2019, February 25). *$44.7 Bn Virtual Reality Market—Global Forecast to 2024*. GlobeNewswire News Room. https://www.globenewswire.com/news-release/2019/02/25/1741284/28124/en/44-7-Bn-Virtual-Reality-Market-Global-Forecast-to-2024.html

Roblox Corporation. (2016). *Roblox*.

Rogers, K., Funke, J., Frommel, J., Stamm, S., & Weber, M. (2019). Exploring Interaction Fidelity in Virtual Reality: Object Manipulation and Whole-Body Movements. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–14. https://doi.org/10.1145/3290605.3300644

Rouse III, R. (2001). *Game design: Theory and practice*. Plano, TX: Wordware Publishing.

Ruberg, B. (2019). *Video games have always been queer*. NYU Press.

RUST LTD. (2016). *Hot Dogs, Horseshoes & Hand Grenades*. RUST LTD.

Salen, K., & Zimmerman, E. (2003). *Rules of Play: Game Design Fundamentals*. MIT Press.

Schell Games. (2020). *Until You Fall*. Schell Games.

Schell Games. (2021). *I Expect You To Die 2*. Schell Games.

Seibert, J., & Shafer, D. M. (2018). Control mapping in virtual reality: Effects on spatial presence and controller naturalness. *Virtual Reality*, *22*(1), 79–88. https://doi.org/10.1007/s10055-017-0316-1

Shelstad, W. J., Smith, D. C., & Chaparro, B. S. (2017). Gaming on the Rift: How Virtual Reality Affects Game User Satisfaction. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *61*(1), 2072–2076. https://doi.org/10.1177/1541931213602001

Shewaga, R., Uribe-Quevedo, A., Kapralos, B., & Alam, F. (2020). A Comparison of Seated and Room-Scale Virtual Reality in a Serious Game for Epidural Preparation. *IEEE Transactions on Emerging Topics in Computing*, *8*(1), 218–232. https://doi.org/10.1109/TETC.2017.2746085

Shneiderman, B., Plaisant, C., Cohen, M. S., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). *Designing the user interface: Strategies for effective human-computer interaction*. Pearson.

Sicart, M. (2011). *The Ethics of Computer Games*. MIT Press.

Statista. (2020). *Oculus VR* (did-30899-1; p. 74). https://www.statista.com/study/30899/oculus-rift-statista-dossier/

Steed, A., Takala, T. M., Archer, D., Lages, W., & Lindeman, R. W. (2021). Directions for 3D User Interface Research from Consumer VR Games. *ArXiv:2106.12633 [Cs]*. http://arxiv.org/abs/2106.12633

Stephenson, N. (2003). *Snow Crash*. Spectra.

Stress Level Zero. (2019). *Boneworks*. Stress Level Zero.

Survios. (2021). *The Walk Dead Onslaught*. Survios.

Sutherland, I. (1965). *The ultimate display*.

Švelch, J. (2016, July 11). Platform Studies, Computational Essentialism, and Magic: The Gathering. *Analog Game Studies*. http://analoggamestudies.org/2016/07/platform-studies-computational-essentialism-and-magic-the-gathering/

Sweetser, P., Rogalewicz, Z., & Li, Q. (2019). Understanding Enjoyment in VR Games with GameFlow. *25th ACM Symposium on Virtual Reality Software and Technology*, 1–2. https://doi.org/10.1145/3359996.3364800

Sweetser, P., & Wyeth, P. (2005). GameFlow: A Model for Evaluating Player Enjoyment in Games. *Comput. Entertain.*, *3*(3), 3–3. https://doi.org/10.1145/1077246.1077253

Tanenbaum, T. J. (2015a). *Identity Transformation and Agency in Digital Narratives and Story Based Games* [PhD Thesis, Simon Fraser University]. http://summit.sfu.ca/item/15285

Tanenbaum, T. J. (2015b). Hermeneutic Inquiry for Digital Games Research. *The Computer Games Journal*, *4*(1), 59–80. https://doi.org/10.1007/s40869-015-0005-9

Tanenbaum, T. J., & Bizzocchi, J. (2009). Rock Band: A Case Study in the Design of Embodied Interface Experience. *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games*, 127–134. https://doi.org/10.1145/1581073.1581093

Tanenbaum, T. J., Hartoonian, N., & Bryan, J. (2020). "How do I make this thing smile?": An Inventory of Expressive Nonverbal Communication in Commercial Social Virtual Reality Platforms. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–13. https://doi.org/10.1145/3313831.3376606

Tanenbaum, T. J., & Tanenbaum, K. (2015). Empathy and Identity in Digital Games: Towards a New Theory of Transformative Play. *Foundations of Digital Games, 2015*. Foundations of Digital Games, 2015, Pacific Grove, CA. http://www.fdg2015.org/papers/fdg2015_paper_49.pdf

Thirdverse. (2019). *SWORDS of GARGANTUA*. Thirdverse.

TiMi Studio Group. (2015). *Honor of Kings*. Tencent Games.

Trammell, A. (2019). Analog games and the digital economy. *Analog Game Studies*, *6*(1).

Trammell, A. (2020). Torture, Play, and the Black Experience. *G| A| M| E Games as Art, Media, Entertainment*, *1*(9).

UpIsNotJump. (2019, May 9). *Blade And Sorcery VR Is An Absolute Nightmare—This Is Why*. https://www.youtube.com/watch?v=V-FuO5k66ro

Urban Wolf Games Inc. (2021). *Legendary Tales*. Urban Wolf Games Inc.

Valve Corporation. (2020). *Half-Life: Alyx*. Valve Corporation.

Vankrupt Games. (2017). *Pavlov VR*. Vankrupt Games.

Violent or graphic content policies—YouTube Help. (n.d.). Retrieved October 21, 2021, from https://support.google.com/youtube/answer/2802008?hl=en#zippy=

VitruviusVR. (2019). *Shadow Legend VR*. VitruviusVR.

VRChat. (2017). *VRChat*.

Ware, C., & Rose, J. (1999). Rotating virtual objects with real handles. *ACM Transactions on Computer-Human Interaction*, *6*(2), 162–180. https://doi.org/10.1145/319091.319102

Wark, M. (2009). *Gamer theory*. Harvard University Press.

WarpFrog. (2018). *Blade and Sorcery*. WarpFrog.

Weech, S., Kenny, S., & Barnett-Cowan, M. (2019). Presence and Cybersickness in Virtual Reality Are Negatively Related: A Review. *Frontiers in Psychology*, *10*. https://doi.org/10.3389/fpsyg.2019.00158

Zhai, S., & Milgram, P. (1993). Human performance evaluation of manipulation schemes in virtual environments. *Proceedings of IEEE Virtual Reality Annual International Symposium*, 155–161. https://doi.org/10.1109/VRAIS.1993.380784

Zhang, H., Ye, Y., Shiratori, T., & Komura, T. (2021). ManipNet: Neural manipulation synthesis with a hand-object spatial representation. *ACM Transactions on Graphics*, *40*(4), 121:1-121:14. https://doi.org/10.1145/3450626.3459830

Zimmerman, E. (2003). Play as research: The iterative design process. *Design Research: Methods and Perspectives*, *2003*, 176–184.